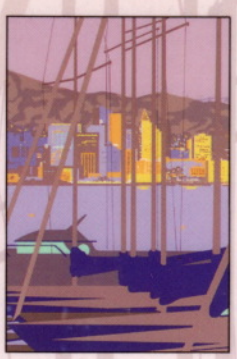
INTEREX



HP Users Conference

August
5-8, 1991
San Diego

PROCEEDINGS

VOLUME 2

MPE, MPE XL, O.A., Networking, Management

Sponsored by

INTEREX

The International Association of Hewlett-Packard Computer Users

Catch the Wave of HP Computing!

INTEREX

The International Association of Hewlett-Packard Computer Users

PROCEEDINGS

of the

1991 INTEREX HP Users Conference

MPE, MPE XL, O.A., Networking, Management

at

San Diego, California

August 5-8, 1991

Volume 2

Paper Number: 4105
NewWave: All About Agents - A Tutorial

Leonard Block
The Apex Group Inc.
7151 Columbia Gateway Drive Suite F
Columbia Maryland, 21046
(301) 290-1606

Introduction

Why NewWave ?? What does NewWave do for me that Windows does not do ? This is usually the first question I always hear when training people on the product. The answer to these questions can be quite lengthy and technical especially for traditional DOS users.

Almost all people are aware of the graphical user interface of NewWave and Windows. Most people who have done some research on NewWave are aware of its strong Object Oriented facility which facilitates easy compound document creation. Some people are aware of NewWave's ability to run other Windows' applications and DOS applications. But only a few people know about the Agent facility in NewWave and even less than that are truly aware of the tremendous capability and functionality that this Agent facility can provide to users.

This paper is intended to be some what of a working tutorial on NewWave Agents. We will start with the basics of the Agent and gradually move into some of the advanced features that the Agent can provide you in your applications development environment. The last section will show how a string of multiple agent functions were constructed to develop a fully automated Executive Information System.

After reading this tutorial, I am sure the next time that you are asked so Why NewWave ? It wont be long before you too are talking about the NewWave Agent as why you chose NewWave to supplement your Windows environment. So let's get started.

Lesson I - What is an Agent Task ?

The HP NewWave Agent automates your work. It can be looked at as a glue that can bind the different objects, tools and individual NewWave, Windows and DOS applications in NewWave together into cohesive application systems and processes. The Agent is really a programming language. It contains most of the capabilities and structures found in other programming languages. One might be surprised to find out that it has a technical reference guide of well ovcr 500 pages.

Almost any function that you perform manually in NewWave such as creating, opening, closing and deleting of objects, copying, cutting, pasting of text, and running applications to name a few, can be automated via an Agent task. These tasks can be as simple as automating one action or can be as complex as stringing together a multitude of actions with no real upward limit.

Agent tasks can be used to automate repetitive and daily tasks which can save users hours of time manually performing the same thing over and over. Besides performing manual options, an Agent Task can also be a complex program. Agents can create customized conversational windows that allow for input of data at run time, display messages, help users decide what branches of a program to execute and so forth.

Like other programming languages, you can tune and enhance your Agent Tasks in the following areas.

* Error trapping and handling
* Interrupt the task and allow user exits
* Alter the speed of running a task
* Control time limits for user responses
* Edit checking and validation of input

These are not all of the capabilities that the Agent can do. The point of Lesson I is that NewWave Agent tasks are processes that cover a broad spectrum in NewWave, from automation of a single function to incorporating multiple tasks and customized procedures into a sophisticated application program.

Lesson II Structure of the Agent Task

So what does a task look like ? In its simplest form the shell of an task would look as follows:

TASK

command
.
.
.
command

ENDTASK

Commands are made up of several kinds of program structures:

Variables, Functions, Procedures, Expressions and Control Structures.

Let's briefly look at these structures.

Variables - A group of characters representing a piece of data in a task. Like other languages, it can assume any given set of values. Unlike other languages, NewWave Variables do not require a data type to be given. Some examples of variables:

folder#="Monthly Reports"	(assign string)
esc#= chr(27)	(assign function of key (escape key)
m#=99	(assign number)
total_sum#=a# + b#	(assign expression)

Functions - NewWave has an extensive set of functions. Each one performs a predefined process. A function is selected based on the process it performs and the type of data value that is returned. Functions work with data either directly supplied or through a variable. Some examples of functions:

GETCLIPBOARDTEXT()	(retrieves data from the Windows clipboard)
PUTCLIPBOARDTEXT()	(places data onto the Windows clipboard)
EDITBOX()	(returns a string containing contents of a text box in a conversational window)
MID(string,integer,integer2)	(Returns a substring extracted from another substring)

The Agent Task Language Reference Guide has over 20 pages of NewWave Functions.

Procedures - Procedures are subroutines. They are separated from the main program body. They are invoked through a statement: **DO 'procedure'**. The use of procedures help organize the task into program modules. Procedures can be executed as often as you wish. Procedures are placed after the ENDTASK statement. An example:

```
TASK
DO INTRO
ENDTASK

PROCEDURE INTRO
FOCUS OFFICE "NewWave Office"
SELECT FOLDER "BEGIN"
OPEN
RETURN
ENDPROC
```

Expressions - Expressions in NewWave allow for the combination of variables, functions and commands. An example might be:

```
month#="January"
select folder month# + " Events"
move_to file_drawer "File Drawer"
```

Control Structures - Control structures allow for controlling the flow of a task. Two main control features exist for you to use in NewWave.

* IF..ENDIF

This structure will allow the task to perform a sequence of events if certain conditions are met. An Example would be:

```
throw_away#="Y"
IF throw_away#="Y"
 FOCUS OFFICE "NewWave Office"
 SELECT TEXTNOTE DOC#
 THROW_AWAY
ENDIF
```

* WHILE..ENDWHILE

This structure will allow a sequence of commands to be executed as long as a particular condition remains true. An example of a WHILE structure:

```
WHILE A# < 100
 DO CALC_RATE
ENDWHILE
```

Being a tutorial we can only brush the surface of the things one can do with the agent. To try and give you a flavor for more than a shell of an agent task, the following small task combines several of the items we have just covered.

```
TASK
'This task shows many different things an agent task can do
'
FOCUS OFFICE "NewWave Office"
INPUT NAME# LENGTH 8 "Please input your name: "
WHILE NAME# = ""
 INPUT NAME# LENGTH 8 "Name can not be blank"
ENDWHILE
IF NAME# = "BLOCK"
 DO INCREASE_SAL
ENDIF
DO FINISH_TASK
ENDTASK

PROCEDURE INCREASE_SAL
NEW_SAL#=OLD_SAL#  + 5000
MESSAGE L# "MR. BLOCK'S NEW SALARY IS " + NEW_SAL# HAND OK
RETURN
ENDPROC

PROCEDURE FINISH_TASK
MESSAGE M# "THIS IS THE END OF THE LINE" YESNO
RETURN
ENDPROC
```

Lesson III Writing Agent Code

The first thing you will notice is that there is an Agent Icon on the desktop when you bring up NewWave. This is the **AGENT TOOL NOT AN instance of an Agent Script Object**. We will talk about the **AGENT ICON TOOL** in Lesson IV (Initiating Tasks). An Agent Script is a NewWave Object. Therefore the first thing that must be done is to create an instance of the Agent object. This is done via the **Create A New** option off the Objects menu in NewWave Office. The best way to begin using Agent scripts is through its built in learn facility or RECORD MODE. Follow these steps and within minutes you will have written your first automated task. The **RECORDING** facility with NewWave Agents will enable you come up to speed on Agents in no time at all while saving you many hours of writing code in your development efforts.

1. Open the Agent object (double click on object)
2. Select the Action menu bar and click on START RECORDING
3. Double click on your waste basket icon on the desktop. The tool will open up and show you if any items are to be discarded.
4. Select the Action menu bar and click on Empty Waste basket.
5. Select the Action menu bar and close the Waste basket.
6. Now go back to your Agent Script, select the Action menu bar and click on STOP RECORDING.
7. Notice the commands that have been generated for you. A copy of these commands can be found in the EMPTY_WASTEBASKET Procedure in the Agent Task at the end of this paper.

Editing

Once you have used the Agent recording facility, this code is now available for customization as if you had typed in all of these commands yourself. The Agent is a programming language and editor built in together. There is no need to use any outside editor. What you see is what you get. The editing is full screen using all the power of Windows editing of data (copy/pasting etc.). You can add, delete, change etc. any agent code you like. This is the process by which you turn simple tasks into complex programs as we talked about earlier.

Compiling

Once editing is finished, an Agent script must be compiled before execution. To compile a script, select the Action Menu bar. Select **COMPILE.** The agent will tell you how many lines are being compiled. If an error occurs you will be notified on the screen and the agent will highlight the line where the syntax error occurred. The line can be corrected right then and there and the task recompiled. Once a clean compile is reached, we are ready to perform our task.

Lesson IV Initiating Agent Tasks

There are many ways to perform agent tasks. We will briefly look at each of the methods available to you. ONLY ONE AGENT TASK MAY BE EXECUTING AT ONE TIME.

1. **PERFORM** - When the agent task is open, one of the actions listed under the Action menu bar is Perform. Clicking will perform the task even though the agent script is open.

2. **DRAGGING** - As mentioned earlier an Agent Icon Tool exists on the desktop. To perform an agent task (object must be closed) drag the Agent script and drop it on the Agent Icon tool.

3. **KEYBOARD** - Click on the Agent script (must be closed) and then type CTRL + F.

4. **Agent Triggers** - This utility (does not come with Standard NewWave) allows for Agent tasks to automatically kicked off upon updating of a predetermined DOS file. The specifics of the trigger are beyond the scope of this tutorial. An agent task can not initiate another agent task. The use of an agent trigger allows this function to happen.

   For example: the last thing Task A will do is export data (convert) from a textnote to a file C:\trigger1. An Agent trigger object is always scanning, checking to see if that DOS file has been modified. If the date and time stamp have been changed the trigger will kick off Task B. which has been previously defined in the trigger object.

5. **Agent Calendars** - This facility allows for the daily, weekly etc. scheduling of repetitive Agent tasks. To see the calendar. Double click on the AGENT ICON TOOL and the calendar will appear. Drop the Agent script on the repetitive scheduling box to schedule tasks. For example, the agent script we recorded above to empty our waste basket could be scheduled to execute every night at 2:00 AM.

Note: A process for Agent Triggers to be activated via voice recognition is currently being developed.

So as you see agent tasks can be performed in a variety of ways. When you are writing, editing and debugging scripts, the first 3 options will be used. As you start to develop complex programs with several agent tasks running in succession, Agent triggers will be invaluable to the success of your system.

Lesson V Agent Interfaces

As I mentioned in the Introduction, the Agent is the glue that binds all the NewWave objects, tools and individual DOS, Windows, and NewWave Applications together. In this lesson we will focus in on some of these interfaces.

Rule #1: Each NewWave object has its own set of Agent Commands. A NewWave object can be anything from a printer (tool) to a word processor textnote (object). Obviously these tools and objects perform different functions warranting that they contain their own set of commands for the agent to execute.

The following is a table of the major object types and the Agent Keyword for them:

Object	Keyword
NewWave Office	OFFICE
File Drawer	FILE_DRAWER
Folder	FOLDER
Waste Basket	WASTE_BASKET
Agent Task	AGENT_TASK
Menu Task	MENU_TASK
Create a New	CREATOR
NewWave Write	NEWWAVE_WRITE
Text Note	TEXTNOTE
Printers	PRINTERS

Each of these objects has many agent commands that can be used in manipulating the specific object. The Agent Task Language Reference Guide has a separate section for each object outlining all the Agent commands and considerations for that object.

Structure of Commands

Usually before any specific agent commands can be imposed upon an object, 3 basic commands must take place. Understanding the sequence of these commands is truly one of the keys to becoming proficient at implementing NewWave Agents. They are:

* SELECT obj_type title_string
* OPEN
* FOCUS obj_type title_string

obj_type = the agent keywords above
title_string= the name given by you to that object

Examples:

SELECT TEXTNOTE "note x"
OPEN
FOCUS TEXTNOTE "note x"

SELECT NEWWAVE_WRITE "DOCUMENT A"
OPEN
FOCUS NEWWAVE_WRITE "DOCUMENT A"

All objects must first be **SELECTED**, then **OPENED**, then **FOCUSED** on.

Since multiple objects can be opened at once on the desktop, the agent needs to know which item to FOCUS on when performing a specific task since each object has its own set of agent commands.

As you write more complex agent tasks with many different object types, one can often encounter many errors due to the agent selecting and focusing incorrectly on objects.

DOS Windows Applications

This area involves Encapsulation (separate topic). From the agent's perspective, the keyword for a DOS object is HPEN_ followed by the name of the object. Use RECORD MODE to help determine the name of the object (Create a new object will give you this information).

NewWave Applications

Any NewWave application that you install will provide you with a new object type that you may create. This object type like all others will have a keyword associated with it that the agent will recognize in its task. Using RECORD MODE to create a new instance of the object is a good means of seeing what the object type keyword is for that object.

Lesson VI Summary

The purpose of this short tutorial hopefully was to make you aware of the powerful agent language that NewWave possesses. It is as functional as many software languages that you find on the market today. It has a lot of characteristics that a 4th generation language possesses. This makes the language easy to learn and understand. Yet it contains many functions that allow for manipulation of object characteristics at a very technical level. You really do not have to be a traditional "C" programmer as is required in many Windows oriented programming applications to reap the benefits of Agents. Create an Agent task object, turn on record mode and you have an Agent task base from which to build on in minutes.

Check the Agent Task Language Reference Guide as previously mentioned for the multitude of things Agents can do. HP is spending a lot of time and effort in enhancing the Agent's capabilities.

Be creative. I have found that by studying the agent functions, one can really develop some dynamic Agent tasks to automate many manual tasks saving countless hours of work while really impressing users as to the power of graphical user interfaces.

As PC hardware becomes faster and cheaper by the day, more and more applications are being downsized or upscaled to a new technology in applications development; **Client/Server** technology. This allows the application to reap both the benefits of mainframe database power and integrity while retaining maximum flexibility that the PC provides. NewWave is an excellent way to manage this new technology. And a good choice to orchestrate this technology is the NewWave Agent which is the glue that can bond it all together.

Lesson VII. A Case study: An Executive Information System via Agent Scripts

The following are some major steps that were used to build an EIS with NewWave as the base. Pieces of Agent code that perform the main functions have been extracted.

Scenario

Every morning a manager wants to see updated sales information from the previous day. Every night the NewWave Agent accesses the host data bases(HP3000 or HP9000) and downloads the transaction data onto the PC. This data is then summarized and exported into a tool "PowerPlay". PowerPlay is a package that allows for definition and viewing of data in various graphical formats (pie charts, bar charts etc.) with a 'Drill down" feature which allows viewing of different levels of data not just the summary format.

Function: Build the PowerPlay carousel for presentations

```
PROCEDURE BUILD_CAROUSEL
FOCUS OFFICE "NewWave Office"
DESELECT_ALL_OBJECTS
CREATE_A_NEW
FOCUS CREATOR "NewWave Office"
CREATE POWERPLAY_CAROUSEL co#+" Presentation"
mess#="        "+"Carousel for "+co#+" Built On "+sysdatetostr()+" at
"+systimetostr()
ENDPROC
```

Function: Define Slides

```
PROCEDURE CREATE_SLIDES
Num_slides#=1
while ns# >= num_slides#
      FOCUS OFFICE "NewWave Office"
      DESELECT_ALL_OBJECTS
      CREATE_A_NEW
      FOCUS CREATOR "NewWave Office"
      CREATE POWERPLAY_Viewer "Slide "+str(num_slides#)
      FOCUS OFFICE "NewWave Office"
      DESELECT_ALL_OBJECTS
      num_slides#=num_slides#+1
Endwhile
FOCUS OFFICE "NewWave Office"
DESELECT_ALL_OBJECTS
FOCUS OFFICE "NewWave Office"
Num_slides#=1


while ns# >= num_slides#
      FOCUS OFFICE "NewWave Office"
      SELECT  POWERPLAY_Viewer "Slide "+str(num_slides#)
      MOVE_TO POWERPLAY_CAROUSEL co#+" Presentation"
      num_slides#=num_slides#+1
Endwhile
FOCUS OFFICE "NewWave Office"
SELECT POWERPLAY_CAROUSEL co#+" Presentation"
MOVE_TO FOLDER "DEMO FOR "+co#
ENDPROC
```

Function: **The Agent Task that follows performs the following procedures:**

INTRO Describes the task

CALC_DB Dynamically determines the name to be given to a PowerPlay EIS database.

FETCH_SERVERS Accesses data from different HP servers(HP3000,HP9000, local PC databases) and converts the data into an ASCII format that PowerPlay can understand for its slide creation. NewWave Access utilizes Client Server architecture.

BUILD_CDB Create the updated EIS data base with the latest data from the servers.

EMPTY_WASTEBASKET Empties the waste basket in preparation for the new day.

```
TASK
EIS_PATH#="C:\HPNWDOS\EIS\"
POWERPLAY_PATH#="C:\PPLAY\"
PARADOX_PATH#="C:\HPNWDOS\PDOXRUN3\"
TRIGGERS_PATH#="C:\HPNWPROG\TRIGGERS\"




DO INTRO
DO CALC_CDB
DO FETCH_SERVERS
DO BUILD_CDB
DO EMPTY_WASTEBASKET
FOCUS FOLDER "DEMO FOR %"
CLOSE
Message m# "All Carousels Updated from Server" OK
ENDTASK
''''''''''''''''''''''''''''''''
PROCEDURE esc_proc
 Message m# "Do you really want to quit?" QUESTION YESNO
 if m# = 1
    Message m# "NewWave Office Demo Factory CANCELLED" HAND
OK
    end
 else
    return
 Endif
ENDPROC
PROCEDURE BUILD_CDB
'
'  Use Creator to Rebuild CDB's
'  Note... this is a bit tricky....
'  First of all the CREATOR under NW does not support Agent,
'  So we can't use normal agent to pass file names....
'  However, we can use a SPECIAL VERSION of the CREATOR
'  Encapsulated as a WINDOWS APP, and pass KEY STROKES!!!
'
FOCUS OFFICE "NewWave Office"

'
' OK Now Use ENCAPSULATED CREATOR to BUILD CDB
'
SELECT HPEN_xxcreator "creator"
OPEN
FOCUS HPEN_xxcreator "creator"
DO_KEYSTROKES "~fn"+cdb#+"~o{ENTER}~fx"
FOCUS OFFICE "NewWave Office"
RETURN
ENDPROC
```

```
PROCEDURE INTRO
Message mess# "This Agent Task refreshes the DeskTop "+ &
   CHR(10)+"by going back to the servers and fetching "+ &
   CHR(10)+"the latest information to drive the slides."+ &
   CHR(10)+CHR(10)+"This Agent Task could be programmed "+ &
   CHR(10)+"to be initiated every night at a pre-set "+ &
   CHR(10)+"time after database updates are done." OK
RETURN
ENDPROC

PROCEDURE CALC_CDB
which_cdb#=1
co#="%"
if len(co#)=0
   Message mess# "Clipboard empty!" EXCLAMATION_POINT OK
   end
Else
   '
   ' Swell... now we need legal characters for a file name
   '
   If len(co#)>7
     hcdb#=left(co#,8)
   Else
     hcdb#=co#
   Endif
   done#=0
   chr#=0
   cdb#=""
   while done#=0
     chr#=chr#+1
     if mid(hcdb#,chr#-1,1)<>" "
        cdb#=cdb#+mid(hcdb#,chr#-1,1)
     Endif
     if chr#=len(hcdb#)
        if len(cdb#) = 8
           cdb#=mid(cdb#,0,7)+str(which_cdb#)
        else
           cdb#=cdb#+str(which_cdb#)
        endif
        done#=1
     Endif
   Endwhile
Endif
return
ENDPROC

PROCEDURE EMPTY_WASTEBASKET
FOCUS OFFICE "NewWave Office"
SELECT WASTE_BASKET "Waste Basket"
OPEN
FOCUS WASTE_BASKET "Waste Basket"
EMPTY
CLOSE
FOCUS OFFICE "NewWave Office"
```

```
RETURN
ENDPROC

PROCEDURE FETCH_SERVERS
FOCUS OFFICE "NewWave Office"
SELECT FOLDER "DEMO FOR %"
OPEN
FOCUS FOLDER "DEMO FOR %"
IF DEMO_TYPE#="3KOnly"
 SELECT NWACCESS "Access HP3000"
endif

IF DEMO_TYPE#="DB2"
 SELECT NWACCESS "Access DB2"
endif

IF DEMO_TYPE#="Both"
 SELECT NWACCESS "Access HP3000"
 DISJOINT_SELECT NWACCESS "Access HP9000"
endif

if demo_type#="9KOnly"
 SELECT NWACCESS "Access HP9000"
endif

IF DEMO_TYPE# <> "PC"
 MOVE_TO OFFICE "NewWave Office"
ENDIF

FOCUS OFFICE "NewWave Office"

IF DEMO_TYPE#="PC"
FOCUS OFFICE "NewWave Office"
CREATE_A_NEW
FOCUS CREATOR "NewWave Office"
CREATE NWACCESS "Refresh PC"
FOCUS OFFICE "NewWave Office"
SELECT NWACCESS "Refresh PC"
OPEN
FOCUS NWACCESS "Refresh PC"
SET_ACTIVE_CONNECTION "PCSERVER"
SHOW_VIEW_DEFINITION
ACTIVATE DATA_VIEW
SHOW_TABLES_LIST
SELECT_TABLE "dBASE.HPDB"
SHOW_COLUMNS_LIST
SELECT_ALL
ADD_TO VIEW_COLUMN 1
REQUEST_DATA
OUTPUT_COLUMN_HEADER OFF
CONVERT_OUT TO POWERPLAY_DAT POWERPLAY_PATH#+CDB#+".DAT"
CREATE
close discard
FOCUS OFFICE "NewWave Office"
```

```
SELECT NWACCESS "Refresh PC"
throw_away
endif

IF DEMO_TYPE#="3KOnly"
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP3000"
 open
 FOCUS NWACCESS "Access HP3000"
 REQUEST_DATA
 OUTPUT_COLUMN_HEADER OFF
 CONVERT_OUT TO POWERPLAY_DAT POWERPLAY_PATH#+CDB#+".DAT"
CREATE
 close discard
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP3000"
 MOVE_TO folder "DEMO FOR %"
 endif

if demo_type#="9KOnly"
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP9000"
 open
 focus NWACCESS "Access HP9000"
 REQUEST_DATA
 OUTPUT_COLUMN_HEADER OFF
 CONVERT_OUT TO POWERPLAY_DAT POWERPLAY_PATH#+CDB#+".DAT"
CREATE
 close discard
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP9000"
 MOVE_TO folder "DEMO FOR %"
endif

if demo_type#="Both"
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP9000"
 open
 focus NWACCESS "Access HP9000"
 REQUEST_DATA
 OUTPUT_COLUMN_HEADER OFF
 CLOSE discard
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP9000"
 MOVE_TO folder "DEMO FOR %"
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP3000"
 open
 FOCUS NWACCESS "Access HP3000"
 REQUEST_DATA
 OUTPUT_COLUMN_HEADER OFF
 close discard
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access HP3000"
```

```
 MOVE_TO folder "DEMO FOR %"
 FOCUS FOLDER "DEMO FOR %"
 SELECT FILE_CONTAINER "Data for Refresh"
 OPEN
 FOCUS FILE_CONTAINER "Data for Refresh"
 DO_KEYSTROKES
"~at{BKSPACE}"+POWERPLAY_PATH#+cdb#+".dat{ENTER}" + &
 "[wait 5]{TAB}{TAB}{TAB}{ENTER}"
 pause 2
 do_keystrokes "~ac"
endif

IF DEMO_TYPE#="DB2"
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access DB2"
 open
 FOCUS NWACCESS "Access DB2"
 REQUEST_DATA
 OUTPUT_COLUMN_HEADER OFF
 CONVERT_OUT TO POWERPLAY_DAT POWERPLAY_PATH#+CDB#+".DAT"
CREATE
 close discard
 FOCUS OFFICE "NewWave Office"
 SELECT NWACCESS "Access DB2"
 MOVE_TO folder "DEMO FOR %"
 endif
RETURN
ENDPROC
```

Paper #4107

Pre-editing Batch Transactions using Reflection, MPEX, STREAMX, and PowerHouse

Presented by:
Steve Hammond
Systems Analyst
Association of American Medical Colleges
1 Dupont Circle, NW
Suite 200
Washington, DC 20036 USA
(202) 828-0448

INTRODUCTION

To understand the process I'm going to describe here, I need to brief you on what we do at the Association of American Medical Colleges (AAMC). The AAMC, among other things, administers the American Medical College Application Service (AMCAS), which one hundred and eight out of the country's 126 medical schools use to process their applications. I am the systems analyst for the AMCAS system.

Although our administrative and support staff have lookup capabilities to the AMCAS and its related databases, virtually none of the data is entered or updated online. Classic 'heads-down' data entry is used for the processing of the data, which is then batched and submitted as part of nightly jobs.

The medical schools receive hard-copy products on their applicant pool daily during peak processing season and twice-weekly in off-season. Weekly, the schools receive via computer tape, or more recently via modem, computer readable data on their applicants. The schools are expected to return, via tape or hard copy, information or 'transactions' concerning their admission decisions on their applicants. In a given processing or application year, we will receive 300,000 to 500,000 of these transactions. The schools are given a predefined format for these files in manuals sent out by the AAMC each spring. This is where the schools asked for change - namely the capability to submit these 'Admission Action' transactions via floppy disk. We chose for security reasons not to allow the schools to upload files to the 3000, so floppy disk appeared to be the best option.

AAMC has a classic (series 70), upon which the AMCAS software runs, and a Spectrum (series 950), with virtually all terminals being HP Vectra's. This allowed us to proceed with our plans to permit transactions to be sent to us on disk. Although such a task sounds very simple, we had to take into account the non-computer orientation of the staff who would actually be handling the incoming disks. As we all know, Murphy is a permanent staff member of most organizations, and AAMC is no exception.

I struck on an interesting mix of Reflection 7 (for the transfer of the files), QUIZ (for the scanning of the files for proper format), QUICK (for the menu to drive the entire process on the 3000) and MPEX and STREAMX (for the actual handling of the files).

Reflection

The command file I wrote for Reflection started out to be so simple. Just a few questions - drive specifications, file name - but the more I thought, the more I realized that several error traps had to be installed to account for any number of possibilities. (I didn't even think to put a 'back-out' option in each step until the first demo to the users.)

The logon was simple enough, that's in all the WRQ manuals. And the prompting for the responses was very straightforward. It was when the time came for processing those responses that I found myself writing more and more code. Granted, I started out as a COBOL programmer, so writing a lot of code is in my blood; I just never really thought it would go this far. Every time I thought I had made this step bullet-proof, I gave it to one of my colleagues and said, 'Break it.' Invariably, it came back broken more often than not.

When it came to that forgotten 'back-out' option, I decided that the best 'out' for each question was a letter 'X'. That would take the user back to an earlier question - or I should say branch in the 'decision tree'. That always gives the user a chance to back out - something which should always be included.

At the start of the command file I issued the escape sequence of '[&k1C', which we all know sets the CAPS LOCK. This will save you a lot of work later on. In a command file that calls for many 'Y' or 'N' responses, having the caps lock set saves you from having to do:

If {Y/N prompt variable} = 'Y' or 'y'...

(Using the SET GLOBAL-CAPS-SCROLL ON command here doesn't work. That merely maintains the condition of the caps and scroll locks when control of the PC is returned to DOS - nothing at all to do with setting the caps lock on.)

After the logon sequence is successfully transmitted, the user is given the option to bypass the file transfer step and go straight to processing the files.

If the user is going to transfer files, a loop starts with a query for drive A: or B: (we had to make provisions for both 5.25" and 3.5" diskettes). After a drive specification, it was necessary to determine the file name for the 3000.

AMCAS has conveniently given each medical school a five-character designation - the first two characters being the state abbreviation and the final three being a numeric code. The user is asked for that school designation which will be used as the file name on the 3000. It fits the naming rules and is easily recognizable and definable (which comes in handy later in the MPEX processes).

I should note here that the user is not prompted for the name of the file in the floppy. The specifications which went out to the schools request that each disk only contain one file and that the file

always be named 'AAMCTRAN'. We provide the schools with labels, clearly showing the school designation, to place on the diskette. If a file on the floppy was misnamed, a separate process on the PC would handle the rename.

Now we've got the school designation, what do we do with it? The command language has nice functions called LIMIT and LENGTH. LIMIT sets a maximum number of characters which can be input for a variable and LENGTH returns the number of characters in the variable. Since a school designation is five characters, I set the LIMIT at five and tested to make sure the LENGTH was not less than five.

But what if they input three letters and two numbers or worse yet, three numbers followed by two letters? The 3000 would kick out that file name awfully fast! With the help of tech support at WRQ I found a nice 'undocumented feature'.

I broke the designation into two pieces, the first two characters and the last three. Here's where the undocumented feature came in. The code went:

```
CONTINUE
LET {new-variable} = LOWER{first-two-characters}
IF ERROR
        {set error flag}...
```

What this basically did was attempt to downshift those first two characters. If either of the characters was a numeric, there would be an error condition. The 'CONTINUE' is just like an MPE 'CONTINUE' telling the processing to continue instead of abort. The same was done with the numeric portion of the designation, except that I added 1 to the variable. If any of the three characters was not numeric, again an error condition existed. The LOWER function does not return an error if the letters are already in lower case. It only returns the error if there are numerics in that field.

So then it was on to the actual transfer of the file. A simple LISTF checked to see if a file by that name was already in the 3000. If it was not, then the transfer was performed. If one did exist, then the user was prompted to either overwrite the existing file or to append to the existing file. The overwrite option was simple, just a 'SEND...PURGE' could be issued.

The append function took some work. When I originally wrote this command file, I was using Reflection 7+, version 3.4. In that command file, I sent the file using the school designation plus an 'X'. Then I ran MERGE.PUB.SYS combining the two files into one, purged the two source files and renamed the merged file back to the school designation. The code went:

```
continue
send {drive-letter} & ":AAMCTRAN" to {file-name} & "X" ascii
if error
        display "^M^J Problems with transfer - try again"
else
        transmit "run merge.pub.sys^M"
        transmit "input " & {file-name} & "," & {file-name} & "X" & "^M"
        transmit "output " & {file-name} & "NEW^M"
        transmit "key1,2^M"
        transmit "end^M"
        transmit "purge " & {file-name} "^M"
        transmit "purge " & {file-name}" & "X^M"
        transmit "rename " & {file-name} & "NEW" & ", " & {file-name} & "^M"
endif
```

Note: '&' concatenates strings together, so that if the file name is 'AL101', the command "purge {AL101} & 'X'" will be transmitted as 'purge AL101X'. '^M' is a carriage return and '^J' is a line feed.

When version 4.0 came out, I saw that it had a new file transfer screen along with an option to APPEND files. Hot dog! No more 'merge ... purge ... purge ... rename' for me!

But not so fast. The append function only works if there is room in the file between the last record and the EOF. And by default, Reflection was creating those transferred files with the EOF being at the last record. It would not be efficient to try to guess how big the file could get and try and set the EOF when doing the initial transfer of a file. Plus the number of times that the append option would be used were so small, it was better the do the 'merge ...purge ...purge ...rename' polka.

But enough about Reflection; let's get onto the 3000 and really do some file manipulation.

PowerHouse - QUICK and QUIZ

The use of QUICK in this project was merely as a menu function. One of the last steps in the Reflection command file was to run QUICK. The QKGO file was already set to run a simple menu screen, with three choices - edit the transferred files, view files which failed the editing and create a tape of all the files which passed the edits. It was decided very early in this project that the data entry staff which would be handling these files on floppy would do no massaging of the data. Simple errors would be corrected by the Computer Services staff and major errors would be returned to the schools for correction.

The COMMAND function in QUICK allowed me to issue file equations and to run MPEX or

STREAMX with input parameters. It was necessary to put some 'PAUSE' commands in the code to hold displays on the screen; otherwise messages just flashed up and were immediately painted over by the menu screen.

QUIZ, in conjunction with some features of MPEX, does the examination of the data. I created a QSCHEMAC which defined the record layout of the transactions submitted by the schools. These layouts are spelled out in documentation presented to the schools and have been the same for over 10 years. The layout is identical to the one used for the tapes they had previously submitted for this process.

The QSCHEMAC defined what I considered to be the crucial items in the transactions which we could edit. Obviously, such a function could not match social security number to name or inspect the validity of a transaction in relation to the existing record on the database (i.e., the action is for an early-decision candidate and the candidate in fact is not on file as requesting early-decision status). What we could check were the items which our operators would see when they dumped the old style tapes we received from the schools to a line printer. Those items were: do the records align properly (you have no idea how many times we would get odd record sizes which would throw the 80-character record out of whack); proper application year; and proper numeric school code (the last three digits of the five-character school designation). Our operators could pick out those errors immediately and would reject the tape. That's just what I wanted QUIZ to do.

I needed to edit all the files in the set of ??### (using HP wildcard definitions) minus GO###. The MPEX function of 'REPEAT...FORFILES' made this possible. Running QUIZ inside of MPEX also allowed me to set file equations for each iteration of the 'REPEAT'. I called the file to be edited 'TRANFILE' and used the MPEX variable 'MPEXCURRENTFILE' to point to the proper file.

QUIZ was then run from within MPEX using MPEX's 'INPUT' statements. These 'INPUT' statements actually were the first four QUIZ commands, which determined the school code to compare to in the record and the school designation to use in display messages. The MPEX and QUIZ code went:

```
repeat
        echo — checking ![lft(rfile.file,5)]
        continue
        file qschemac=flopschc
        file discrpt;dev=disc;acc=append
        file tranfile=!mpexcurrentfile
        run quiz.current.cognos;stdlist=$null;      &
                input="access TRANFILE";&
                input="def SCHOOL num*3 = ![rht(rfile.file,3)]";&
                input="def FNAME char*8 = '![lft(rfile.file,5)]'";&
                input="use RPTUSE nol"
forfiles ??### - GO###
```

Note: 'rfile.file' is the name of the current file in the REPEAT loop.

The last 'INPUT' statement is a QUIZ 'USE' command which executes the rest of the QUIZ code. The caveat on the 'INPUT' statements is that there is a limit of 256 characters which can be used in the

statements for any given 'RUN' command. Otherwise I might have done the whole thing with 'INPUT' statements.

The 'USE' statement defines several variables as error flags and determines the application year. (Each application year is a different account, i.e., AMCAS90 or AMCAS91. Using the QUIZ variable 'SIGNONACCOUNT' I was able to do a bit extraction and convert the last two characters to a numeric variable which was the application year.) The school code (the three-digit numeric segment of the school designation) is passed to the QUIZ processing with a DEFINE statement in the INPUT statements following the RUN statement out of MPEX. I used the MPEX bit extraction to retrieve the code from the file name (another reason I used the school designation as the file name).

A file is then read completely and if errors exist, error flags are set. A temporary QUIZ subfile is created, sorted and a report is sent to a disc file. This report file was built prior to the QUIZ run and is set with the 'ACC=APPEND' option. (Be careful of files with this 'ACC=APPEND' option. Ever try to read one of those files in TDP? You get one of those errors which is described in the manual as, 'You will never get this error, but...'. I always reset the file to 'ACC=INOUT' when I'm done with it just to be safe.) This disc file called 'DISCRPT' will have one line of text for each file processed. The line will say 'No errors in data from [school designation]' or 'Bad School Code in ...' or 'Invalid AMCAS year in ...' or 'Bad transaction format in ...' or 'Bad School Code and Invalid AMCAS year in ... '. The file looked like:

Bad School Code in AL101
Invalid AMCAS year in MD133
No errors in data from TX804
Bad transaction format in OH156
No errors in data from FL806
Bad School Code and Invalid AMCAS year in NM150

Remember this disc file because we'll be looking at it later in the MPEX portion of this presentation. Suffice it to say, this little file is the heart of the remainder of the processing.

I use QUIZ at one other point in the processing basically to create a line of code for STREAMX to execute. The QUIZ run retrieves a value from a record in the AMCAS database. It creates a variable with a 'SETVAR' command using the value retrieved as the variable. This one line, 24-character statement is written to a disc file. STREAMX then "XEQ's" that file to set a variable for subsequent use in the STREAMX run.

Basically I use QUIZ to create just two little files, but they are crucial to subsequent processing.

MPEX

So now we've checked the files and we know 'which have been naughty and which have been nice'. What do we do from here? Basically we get into some very heavy MPEX and STREAMX processing.

I should note here that every time I ran MPEX in all of these processes, I ran it with a 'PARM=1'. This is the fast-start mode for MPEX. The process does not need the user's UDC's loaded or need to maintain the REDO command history which MPEX will normally do at startup time. This cuts down on considerable overhead and gets things going a bit quicker.

To back up slightly, the QUIZ process mentioned a moment ago was run out of MPEX, using the 'REPEAT...FORFILES' command. All of the files in the set ??### (minus any GO###) were processed and the results of that processing were put into a disc file. I then used MPEX to examine that disc file called 'DISCRPT'.

On version 2.2 of MPEX, it was necessary to create a son process to have MPEX examine a file. It was a matter of running MPEX from within MPEX and for that son process, stdlist was set to $null and stdin was set to the file to be examined (in this case DISCRPT). Additionally, the son process had to be run with an 'INFO=xxxx' parameter, with 'xxxx' being the MPEX execute file needed to perform the steps within MPEX.

I couldn't wait to get my hands on version 2.3, because of a little function called VEFOPEN, which will open a file directly from MPEX and allow you to read a record using VEFREAD functions. If you put that read inside a 'WHILE...' loop, you can read the file until the EOF. That code was:

```
setvar numrecs vefinfo("DISCRPT").eof
setvar filenum vefopen("DISCRPT,old")
setvar K 1
setvar J 0
while K < = numrecs
  setvar dataline vefread(filenum)
  if dataline matches "@No errors@"
     :rename ![dataline[23:5]], GO![dataline[25:3]]
     setvar J J+1
  endif
  setvar K K+1
endwhile
```

Here the code looks complex, but really only does a few things. It gets the eof of DISCRPT with the VEFINFO intrinsic. Then, using VEFOPEN, DISCRPT is opened for reading. A variable for counting lines (K) and a variable for counting the number of renames (for a later ECHO), are set to 0 and 1 respectively. Then a WHILE loop is begun. As long as the number of lines read (K) is less then the eof, the WHILE continues. In each iteration, the next record in the file is read into the variable DATALINE. If the string 'No errors' appears anywhere in DATALINE, the file is renamed from its existing name of state code plus school code to 'GO' plus school code; i.e. AL101 becomes GO101. (Remember QUIZ

placed the literal 'No errors in data...', so that was the target to match on.) This is done with a set of bit extracts from DATALINE. The position of those values in dataline remains constant since they are written by the same QUIZ routine. (I chose 'GO' because no state has 'GO' as an abbreviation and it would make the files easy to identify if there was a problem.) I actually did an MPE rename instead of the MPEX rename (that's done just by putting a ':' in front of the command). That was done to avoid confusion to the user. The MPEX RENAME gives a fancy display with names and arrows. That would only lead to phone calls from the user. Besides, an MPE RENAME avoids the added processing which MPEX does before the execution of the command.

After the files are renamed, I use the PRINT command to display DISCRPT to the screen. DISCRPT contains the names of the files and what edits they failed or whether they passed everything. This is the first time the user knows which files will be passed on for updating the AMCAS database and which will either need to be massaged or rejected completely. The statement for the print was:

print discrpt;search=(recnum mod 3)=0

This command prints every third line of DISCRPT, because even with the access being appended, QUIZ would output two blank lines prior to the message. This display is followed by an ECHO of the number of files which were processed and the number which passed the edits. All this will help the user track problem schools and how many files will be processed on a given evening.

So the target files have been examined and renamed. But what was wrong with the ones which failed the edits? The user has to find out. Maybe the user gave a wrong file name way back when they copied the file from the floppy to the 3000. Maybe they put it in one application year's account and it should have been in another. Maybe the medical school sent us a Lotus 1-2-3 file instead of an ASCII file in our record layout. Maybe the medical school sent us a formatted report of all their admission actions out of DBase. (Don't laugh, both these things happened in early stages of this project.)

Basically, the user needs to know why these files were rejected. The user did get an indication from that little QUIZ file if the school codes were bad or the application year was bad or the format was bad. But in some cases, it could have been an error by the data entry staff which upon examination could be corrected easily.

So I displayed file to the user, by using the MPEX 'PRINT' command. Throw 15 lines of the file on the screen and the user ought to be able to find the problem. As simple as it sounds, I have to say this is the most difficult section of the code which I had to write - 48 lines (less comments) of a debugger's nightmare.

As an aside I think I should mention something here about comments. I know the old adage 'Real programmers don't comment code. If it was hard to write it should be hard to understand.' That's all well and good, but those comments helped me when I went back to tweak the code, and they helped even more when it came time for me to write this paper. I printed out all the code for this project when I started this paper and I found that 45 percent of all the lines of code written were comments. It made this paper much easier to write, and it made updating the code much simpler. I'm reminded of the grieving family at the graveside, with a man whispering in the widow's ear, 'This may be a bad time, but did he ever mention anything about source code?' Remember that!

The reason this print function was so difficult centered around the traps I had to put in for 'Do you want to continue' and 'Invalid file names'. Suffice it to say that when you are accepting multiple 'Y/N' responses from a user in a process you can get your hands crossed very quickly.

One nice feature I did discover was that I could edit the response for the file name at entry time. Again it was the 'IF...MATCHES' function from MPEX. It was a simple 'MATCHES ??###' to check the validity of the format of the file name. The code was:

```
setvar filenm ""
input filenm;prompt="Which file do you want to see?"
if filenm matches "??###"
        {continue}
else
        {{error routine and loop-back}}
```

This function is detailed in the appendix of the MPEX manual. Those of you who have never looked at it should take some time and study it. Understanding the functions detailed in the MPEX appendix made my work with MPEX far more productive.

Once we got the file name, it was a simple matter of placing a line on the screen which was like a TDP 'rack' or a QEDIT 'list template', showing space numbers and then 'PRINT' the file's first 15 lines. The problems I had came in trapping and checking the right responses for continuing with more files, seeing more of the displayed file, etc. We're talking a day of coding here and several calls to tech support at VESOFT to get it right.

STREAMX

Do you know you don't have to use STREAMX for just streaming a job? I knew about all these nifty things you could do aside from avoiding passwords in job streams. You know, pre-submission processing, customized TELLs to the sender. Have any of you still on a Classic noticed that MPE V now puts the tombstone at the top of the stdlist showing who streamed the job - just like STREAMX has been doing? I wonder when that will make it into the XL operating system?

But STREAMX does not have to be used to stream a job. You can create a file of nothing but those pre-submission processing commands, which use a '::' to denote their intention. I built a file of nothing but these commands and used them to process the final step in this project. One other nice thing is that MPEX variables are session-intensive, meaning that an MPEX variable set earlier in this session can be referenced by the subsequent STREAMX command file.

I'm getting ahead of myself. Basically, I have any number of files with the naming style of 'GO###'. These files need to be merged into one file and copied onto a tape. The tape will be used as input to the nightly processing of AMCAS, and the parm options following the RUN statement indicate how many tapes will be used as input for the run. Both time and logistics called for the jcl to be updated to reflect an additional tape by this process, rather than having the user call the computer operations staff to ensure the jcl is updated. Also this step had to be separated from the editing step, because the former

could be done several times a day, while the latter would only be done once.

On version 2.2 of STREAMX, I had to run three different son processes to use MPEX to set some variables. Version 2.3 allows MPEX commands to be executed directly from STREAMX (using a '::%' prefix), so all of those son processes were eliminated. I still needed a 10-line QUIZ routine to retrieve a variable from a database at the start of this step.

Basically, I opened the jcl file (using that helpful VEFOPEN command again) and read through the file until I found the parm card. I then did a bit extraction to find the number of tapes listed for use in the run for that night. I stored as variables the existing number of tapes and the line upon which the parm card fell, and computed the existing number of tapes plus one. The code was:

```
setvar update_jcl_eof vefinfo("RUN2JCL.JOBS").eof
setvar jclfile vefopen("RUN2JCL.JOBS,old")
setvar line_ctr 1
while line_ctr < = update_jcl_eof
        setvar dataline vefread(jclfile)
        if dataline matches "!RUN UPDATE@"
                setvar dataline vefread(jclfile)
                setvar tape_cycle_num_jcl ![dataline[13:3]]
                setvar tape_old ![[dataline[46:1]]
                setvar tape_new tape_old+1
                setvar line_found line_ctr+1
                setvar line_ctr update_jcl_eof+1
        endif
setvar line_ctr line_ctr+1
endwhile
```

This code does the same loop as in an earlier routine, setting a variable to the eof of the file and then doing the WHILE on the line counter being less than the eof. Each line is read into the variable DATALINE. If the line begins '!RUN UPDATE', then the next line, which is the parm card, is read and several values are stored or computed, such as the line number where the parm card was found (used in the TDP CHANGE command later), the old number of tapes and the new number of tapes (old number plus 1), and the cycle number of the next update run. Then the line counter is set to one higher than the eof to end the WHILE loop (no sense reading anymore since the parm card has been found).

I also checked to make sure the jcl had been updated that day by looking at a value on the parm card. (If the jcl hadn't been updated I stopped the step immediately to avoid overwriting of the file after the changes had been made.) Then I needed to build the file into which all these 'GO' files will be copied. Here I executed an MPEX command file. You can issue MPEX commands from STREAMX as long as the actual command does not do any file-set handling. Well, the command file does a 'REPEAT...FORFILES GO###' - in other words, file-set handling. But the command in STREAMX is a simple '%XEQ...'. I let the command file do the file-set handling and all STREAMX saw was the

XEQ command. This routine just added up the eof on all the GO files and then added 100 (just to be safe) to yield yet another variable. The code was:

STREAMX command was **::%xeq totalrtn**
MPEX XEQ file was

```
listf go### > $null
setvar num_file mpexnumsucceeded
setvar tape_tot
repeat
        setvar num_recs vefinfo("!mpexcurrentfile").eof
        setvar tape_tot tape_tot + num_recs
forfiles go###
setvar tape_tot_plus tape_tot + 100
```

I ran into a stack limit here which caused some consternation. I was able to get around it by setting the JCW STREAMXVARAREASIZE down to 1000. That size is defaulted to 4096, but you can reduce it by setting the jcw. You need large stack size if you are handling a lot of string value accepts. I think I blew the stack because I had so many variables, and I could reduce the area in the stack for string values because I was using so few. Anyway, this worked.

A file to hold these merged files was then built using the variable which was the total of the eof's plus 100 as the number of records. I then set the access to append and issued another MPEX execute command to copy the files to the merged file. The MPEX FCOPY command was used, as opposed to COPY, because COPY would not handle the appending. (It kept asking me about purging the existing file.)

Now we've got all the 'GO' files merged into one file called TAPEFILE (clever, huh?) on disc, and it's just a matter of copying that file to tape. Here's where I got together with the users and the computer operations staff to make some decisions. What was decided was that the actual step to update the jcl and create the tape needed to be in a job stream as opposed to this interactive processing. That way, there would be a paper trail to allow late shift operators to check if the tape had been created and the jcl was up to date. In this electronic age, it's still nice to have that piece of paper to fall back on.

I just ran STREAMX out of STREAMX with a PARM=1 and the STRMFILE set to the proper job stream file name. This last job was another job filled with STREAMX commands along with MPEX thrown in, but basically it checked the day of the week to determine which tape to have the file copied to (we recycle AMCAS input tapes every week). It also used the EDITTDP command to change the number of reels from x to x+1 on a particular line (that's why all those variables were set!!) The MPEX command to change the file was:

```
EDITTDP RUN2JCL.JOBS,&
        c "REELS{tape_old}","REELS{tape_new}",lit,{line_found};set yes
```

It then copied TAPEFILE to tape, and the process was completed. That wasn't so bad was it?

While STREAMX is doing all of this, it's quite important to do periodic ECHOs back to the user. This takes a lot of time, and the user is just going to sit there staring at a blank screen if you don't

enlighten them occasionally. Five different times, I check the status of job and inform the user - if there are no GO files, I stop the process and I tell them. If there are more than nine tapes in the existing jcl, there's no way we can handle 10 (we've never had more than five input tapes in my five years of overseeing AMCAS). Silly little things like that need to be checked so the user can be appraised of the process. The only time the computer services staff should ever become involved in this entire process is when an operator hangs a tape. I realize that's a dream and not reality, but in this era of distributed processing, the more the user can do, the better it is for the techies!

EOJ

This project started out so small, but grew as we all realized its significance. It could free up data entry and computer operating staff by moving the processing of Admission Action transactions (of which we will receive in excess of 300,000 a year) onto a PC and back even further to the PCs at the medical schools' admission offices (who claim to be the last to get any computer time at major education institutions). I was able to use some powerful tools including Reflection, MPEX and STREAMX to get it done, with the minimum of impact on the user and still meet the growing demands of the medical schools' admissions offices.

At my old job, I kind of fell into the job of systems manager when the old one quit. At one point, he said to me, 'MPEX is your friend!' I've never doubted that, and I think this project proves that Reflection, STREAMX, and PowerHouse are also some of an HP3000 user's best buddies.

Acknowledgements

Several individuals helped with this project and the preparation of this paper and I would like to thank them: from the tech support department of Walker, Richer and Quinn, **Bruce Marshall**; from the tech support staff of VESOFT, **Tim Joseph** (now with Bradmark), **Gil Milbauer**, **Michael Hensley**, **Tom Emerson** and, of course, **Eugene Volokh** and **Vladimir Volokh**; friends and co-workers **Alan Ambers** and **Laurie Ambers**. I am also grateful to my supervisors, **Brendan Cassidy**, **Sandy Lehman**, and **Maryn Goodson** for letting me run with this project, when they could have made me do something far more important, but far less interesting.

Notes

MPEX and **STREAMX** are products of **VESOFT**. **PowerHouse**, with its components **QUICK** and **QUIZ**, is a product of **Cognos**. **Reflection** is a product of **Walker, Richer and Quinn**. The following versions of software were used in this project: Reflection 7+, versions 3.4 and 4.0; PowerHouse, version 5.06; and MPEX and STREAMX, versions 2.2 and 2.3.

Paper# - 4108 Implementing HP's NewWave Office by Dr. David Johnson

Johnson Computer Software Team Limited, Suite 5012, 3080 Yonge Street, Toronto, Ontario, Canada M4N 3N9 (416) 487-3631

JCST is a software development company specializing in HP NewWave and cooperative computing. The company also does a certain amount of related facilities management. The following is an account of the planning, development and implementation of HPs NewWave Office at one of our major clients

Implementing a NewWave Office environment to replace a traditional on-line system is, in many ways, as dramatic a change as it was to go from batch to on-line. For the end users, of course, it is a revolutionary change but for the technical people too it can be a surprisingly dramatic change.

We learned a number of lessons, some the hard way and some, fortunately, less painfully. Most importantly:

- *When preparing to implement new technology try to persue the vision through to the last details. It is the details that can wreck such a project.*
- *Don't rely on the traditional source of information about products - the manufacturer - to be able to tell you anything about the products - ultimately the only thing you can rely on is your own experience. On the other hand there are still some 'old faithfuls' you can rely on. It's not always easy to tell who your friends really are but try in advance and be sure to line up all the support you can. The answers are wherever you can find them.*
- *New and to some extent revolutionary technology requires time and effort to assimilate and tune - there's a lot of work to do but there are real rewards.*

It all started when the old series 42, while not exactly on it's last legs, was overworked and starting its 5 year obsolesence. The full backup often went to a ninth cartridge tape, one more than the 35401 cartridge magazine would hold. This meant users could not always get on to the system first thing Monday morning. The wonderful little HP150s had been declared obsolete about the same time as the series 42 and the main application run on them - HPWord - was declared "mature". That was 1988 and it was time to move on.

At the same time a few PCs had, due to various necessities, made their way into the organization so they were not totally foreign. We had, for example, learned that they were *not* like the HP150s that you just unpacked, plugged in, turned on

and away you went. Even ordering PCs so that you got all the pieces necessary to make them run was a challenge in those days. It's easier now they come "bundled", but this was a harbinger of things to come.

The first thing to do was to take stock of where we were. A quick inventory of the hardware:

- a series 42 with the maximum memory, close to the maximum number of ports and an INP for DS communication to other systems
- about 800 Mbytes of disc spread over discs of varying vintages
- a 1600 BPI tape, a cartridge magazine and a line printer
- a host of HP150s (connected serially to the 42) used as word processing stations and terminals

This inventory revealed very little with inherent value worth trying to save. In fact only the printer and one disc drive would survive the migration. The good news was that it was all wearing out so to speak - and written off - at about the same time.

The software inventory was likewise very straight forward:

- HPWord running as shared processing on the series 42 and the HP150s
- HPDesk on the HP3000
- custom applications on the HP3000 using IMAGE databases and the HP150s as terminals
- applications to link HPDesk to other E-mail systems
- DS to connect to other HP3000s
- various stand alone PC applications on both the HP150s and the few Vectras

Examining what we felt were emerging requirements within the organization several factors emerged:

- the importance of a common user interface: users who were familiar with the custom applications where the f7 key was *always* Refresh were baffled when, using HPDesk for the first time, they found that f7 did not mean Refresh. There

was a significant learning curve associated with implementing software from different vendors.

- the facility for context switching between applications: more and more secretaries were finding it necessary to switch between HPWord, Lotus and HPDesk for example. We had implemented Execudesk on several HP150s - we were warned off Windows for the 150. But because it was running in RAM it was very sensitive and took a fair amount of time to bring up. Moreover it was not compatible with HPWord which required the entire HP150 - loading Execudesk destroyed HPWord and vice-versa and it took several minutes (even at 19200 baud) to reload HPWord.
- the necessity to remove the complexity of "integrating" applications: it was, for example, a major chore to receive a Lotus spreadsheet in HPDesk and get it down to the PC and into Lotus. It's a cliche by now, but users should be able to concentrate on doing their jobs not on the computer techniques necessary to accomplish them.
- the necessity to put computer power on users desks: shared processing on a mini put users at the mercy of other users, system backups etc. where they had no control or even direct knowledge of why response time was bad or the system unavailable.
- the necessity to free users from the responsibility of the integrity of their own systems: while giving users computer power on their desks, those desktop machines needed to be connected to a server, and the users should not have to be responsible for that connectivity or for their own backups etc.

A quick tally of the functionality that was required produced the following list:

- word processing (of course)
- spreadsheets
- graphics
- electronic mail both internally and connected to outside mail systems
- various custom applications unique to the organization
- connectivity to outside systems (both to other HP3000s and to various dialup services)
- a storage and retrieval system to replace a cataloguing system built on the series 42 for HPWord documents

Now came the more creative part of defining where we wanted to go. Information technology must be a major strategic component of any corporations planning. It's not enough just to upgrade individual pieces of hardware as they become obsolete or reach their capacity. It's necessary to have a comprehensive vision that must be shared and shaped at the most senior levels. It's the sort of planning that answers the question "where do we want to be in 1995?".

We were very excited by the announcement of NewWave in Dec 1987. Having ordered a developers kit we discovered first hand in early 1988 the power and advantage of the Windows interface and began to explore the capabilities and possibilities of NewWave. To a very large degree the vision was shaped by our understanding of NewWave.

The vision became fairly easy to articulate: PCs on users desks with all interaction to the system being through the NewWave paradigm.

Why PCs? The choice was among PCs, Macintosh or workstations. Workstations were not given serious consideration because they were too expensive and very little commercial software was available. The Macs were nifty machines but PCs were already entrenched so two diverse types of hardware would have to be supported and integrated (a very important factor as it turned out) and it was not clear they would integrate with the HP3000 applications in any way except terminal emulation. We definitely wanted to progress beyond terminal emulation.

Why NewWave? We were sold on the Windows interface but not Windows itself. The Windows interface was intuitive and it was standard (the success of Windows 3.0 has only made it more so). But Windows still requires the user to understand DOS files and directories. The user must understand he is running a program and has to load the right file(s). The object motif of NewWave vastly simplifies it for the user. The additional features and facilities of NewWave such as the CBT, the ubiquitous help and especially the Agent made it very appealing. Besides, if HP had invested so much (I'm told it was the second largest R&D project ever done second only to the Precision Architecture systems) it would be around to stay. Wouldn't it? With it's encapsulation facility NewWave provided a very appealing environment to integrate all the various applications in use or required whether or not they were native NewWave.

The investment in custom software on the HP3000 meant an HP3000 would be part of the solution in any case. With the emerging industry consciousness of the importance of "standards" we took a second look at staying with a proprietary operating system but the cost of doing anything else would have been prohibitive. Besides MPE will "look like UNIX" sooner or later so maybe we have a migration path to an industry standard (sometime after HP deems UNIX fit for "mission critical" applications). In the meantime we kept it in the back of our minds in all subsequent decisions that we wanted to eventually reduce and even eliminate our dependance on MPE.

With PCs planned for the users' desks and an HP3000 in the computer room the next component was a LAN. Again with the drive to standards in mind we paid heed to the rumours that the same group that wrote LAN Manager for HP-UX machines was now writing it for the HP3000. Maybe God was in his heaven and all was right in the drive to standards. In the meantime Officeshare was there as an interim step and HP has, of course, the reputation for protecting their customers and providing necessary upgrade paths (yes they do and they did).

The next step was to look for NewWave, or at least Windows, software to provide the necessary functionality. Windows software that had a NewWave future was definitely given priority over straight Windows software which in turn was given priority over non-Windows software. We learned, however, that this was not the only criterion to apply.

We were very, perhaps unusually fortunate, that we were not under serious time constraints when we embarked on the project. We could take all the time to plan, evaluate and develop that was necessary (late, external considerations forced the final conversion on us). We were also fortunate that the project was "owned" by senior management who shared and shaped the vision. Recommended reading for anyone and everyone is the MIT study **Management in the 1990s.**

Having pretty much decided that NewWave was our "shell", we started evaluating software to fit under this shell.

- AmiPro from Samna got the nod over Word for Windows because, while they both did all the things a GUI word processor should do, AmiPro was destined to become (and now is) a NewWave application. It's very convenient to be able to select several large documents that need to be printed, drag them to the printer and go for lunch rather than having to open up each one, tell it to print, wait for

it to format then close it up and go on to the next. We had originally planned to use either Ami or NewWave Write for simple documents and letters and limit the more powerful and complex AmiPro to a few power users - secretaries - and not have to bother training executives on AmiPro. This is *not* the way to go. In the NewWave environment with the ability to pass objects around it is important that everyone is using the same system.

- With the existing investment in HPDesk, NewWave mail looked like a natural so after months of using the evaluation copy this was chosen.

- When we got to the issue of spreadsheets we learned first hand the old lesson that "best isn't necessarily better". Excel looked to be the obvious winner, it was destined to become a NewWave application according to Microsoft and already had several adherents within the organization. The investment in Lotus turned out to be too great to abandon. It was just too expensive in terms of retraining and converting existing work sheets to switch to Excel. The good news was that HP had done such a good job of integrating Lotus into NewWave that we could live with both without losing very much of the vaunted NewWave functionality. Then the good news was that Lotus was going Windows and "seriously considering NewWave". At the time of writing, Excel 3.0 has shown up with a lot of NewWave functionality although it is still an encapsulated Windows application. Lotus has released it's Window-like version 3.1 but no sign of the real Windows version yet. Maybe they will gain some advantage in this from their acquisition of Samna, the people who brought us AmiPro.

- When we got to the issue of graphics we learned that just because it is in the New Wave catalog doesn't mean it is or ever will be a NewWave application. Graphics Gallery is still being used and, to date, nothing else has been found to fully replace it. Once again, because HP has integrated the Graphics Gallery into NewWave, there is no urgency to find a replacement.

- There was no obvious functional replacement for the cataloguing system we had developed on the HP3000. In it, HPWord documents were scanned nightly by a batch job and entries in a catalogue database updated so users could subsequently locate documents based on various attributes. HP had hinted of something of this nature being available through HPDesk but, in the first hint of trouble to come, no subsequent information - for or against - was ever forthcoming. This would have to be developed. What more natural to the NewWave metaphor than an Archive - as it would be called - where the user dragged objects to be stored and where they would subsequently go, trusting only to their mouse skills, to locate objects: any NewWave objects, not just word processing documents.

- The custom applications would have to survive in terminal emulation mode until such time as they could be suitably converted to the NewWave office metaphor.

We had developed expertise putting a Windows interface on HP3000 applications but this was unsatisfactory when trying to convert the user's view of the system to the NewWave interface; better ways have subsequently been found. The first terminal emulator on the market was Session from Tymlabs. We bought it and worked with them for months and months on debugging it. Why we switched to R1WIN from WRQ is a later part of the story.

By late 1989 we had figured out all the pieces and they looked like they would fit together. Oh sure, we had memory problems with Windows 2.11, but these were generally when a lot of context switching was being done. You could get Windows in such a knot that even though all applications were shut down, if you tried to bring up the "About" box to see how much memory Windows thought it had Windows told you it didn't have enough memory. Not to worry, the end users didn't stress the system the way we were doing. The Agent wasn't implemented but, as a developer, we had a release of NewWave with Agent facilities - it would be along soon. We got one shock when we discovered AdvancePrint - which we had been assured would work - didn't. Moreover we were informed (by the Response Centre) that it was not going to work with NewWave. Not to worry, the required functionality would be available from Resource Sharing on the LAN. Now the only consideration was the hardware: the smallest Precision Architecture machine was a 925. It was too large in size and cost for what we needed but, not to worry, smaller machines were rumoured to be imminent, and they were. HP announced the 922s in a server version. With a 922, by eschewing terminal connectivity, you get the hardware and software for connecting the PCs on a LAN for no extra cost. Just exactly what we wanted. God was in his heaven....

So we sat down with HP to order the pieces. Well what pieces? To this day I'm still not sure what pieces you need for NewWave Office nor do the people at HP. In all fairness to the HP sales rep who spent a lot of time on the 'phone to California, he didn't have the requisite technical support to ever give us a fighting chance. When he was transferred to another office between order and delivery and a new rep assigned, it was pretty well guaranteed that we would never get it sorted out beforehand. We knew, from working with other clients, how to go over an order for a classic HP3000 and make sure everything was just as it should be (down to the last Y-cable) but we had no experience with PA machines or with LANs or NS for that matter. We were totally reliant upon HP to tell us what we

needed and to get it ordered. We also played it safe and ordered FastLane, NetStartup and NetAssure to go along with the Response Centre support already in place. Let me hint at what is to come and state right now that the Response Centre is like American Express: don't leave home without it. In fact it is safe to say that without the Response Centre the project *would very likely have failed!* My heartfelt thanks goes out to the people at the Response Centre who worked so patiently with us through our problems.

So the hardware and software were ordered.

We learned another valuable lesson the easy way. A couple of very similar experiences taught us, long before implementation time, just how different NewWave was going to be for end users. In one case the very PC literate project owner in the organization had NewWave on his PC for the first time and asked "it looks very nice, what do I do with it?". Oh. We had been using NewWave in the form of the developer kit for some time so that it all seemed so intuitively obvious to us as to obviate the need for any explanations. In another very similar instance, a very bright, very articulate MIS person in another organization related to us that he had a loaner PC with NewWave and didn't know what to do with it. Obviously this was not something we were going to have any success with if we just dropped it on a secretary's desk.

Training - formal training - was going to be necessary. HP had sent a brochure about all the wonderful training they could do and, moreover, if you didn't see what you wanted just call them and design your own course. Suffice to say we ended up designing and conducting our own user introduction to NewWave. HP does have a course but it's for administrators. We, who have three years experience designing and developing Windows and NewWave applications, have no one individual that can meet HPs official prerequisites for attending the course. It's definitely not the sort of thing for a secretary to learn about NewWave.

Fortunately, NewWave comes with built in tutorials - this is the CBT feature - that uses NewWave to help you learn how to use NewWave. The tutorials are well designed and very cleverly put together and served as the basis for our introductory course. In fact, as an indication of how much work went into them the Agent tasks that are used to drive the lessons are, we are told, too big to be contained and compiled in a normal Agent object. Special unsupported techniques had to be used to compile them. A result of this is that the Agent task

source language for these lessons is not available on the system (the lesson objects and containers are invisible on the end user system but are visible on the developer version). It would have saved a lot of time and frustration when developing our own CBT lessons to be able to see how HP did certain things. Suffice to say the lessons do what they are supposed to do and do it exceptionally well. *Anybody* can conquer NewWave in a half a day and this includes computer illiterate executives, 8 year old kids, retired school teachers and secretaries who, when first told to move the mouse, picked it up and waved it around in the air!

A half day course, out of the office and away from distractions, was sufficient to overcome almost all new users trepidation. In many cases just this amount of exposure was sufficient to generate real enthusiasm for the possibilities that were being opened up.

But before any of this training started we waited for the system to arrive. Delay after delay was evident as, we were told the (server) software wasn't ready. Finally, months late, the 922 arrived. We still didn't have enough software available to do anything more than connect the console, we were still waiting for software (it was at our insistence that HP deliver the hardware so we could at least start to do *something*).

To our surprise and puzzlement, contained in some of the myriad of boxes, were OS/2 and LAN Manager! We had, early on in the ordering process, enquired as to the availability of LAN Manager since, all other things being equal, it is preferable to go with an "industry standard" solution rather than a proprietary solution, and were told it was not yet available on the HP3000. When it became available an upgrade path from Officeshare would also be available (one does admittedly take this latter for granted when dealing with HP). Seems this somehow translated into an order for LAN Manager on the PCs and Officeshare on the host.

So what were we to do with LAN Manager? Not to worry, LAN Manager on the PCs is backward compatible with Officeshare on the HP3000. Because we didn't understand what pieces were what in the networking environment we questioned the wisdom of trying to mix LAN Manager and Officeshare. Should we be using Officeshare on both ends? HP came in with Officeshare discs and some advice: if we retreated to Officeshare there would be an upgrade charge if and when we

eventually moved to LAN Manager (such a move would presumably be required eventually as HP moved to open systems and phased out proprietary software such as Officeshare). But, and this is significant, we were also advised in no uncertain terms that if we planned to move our PCs to the recently released (4 months earlier) Windows 3.0, the drivers for Officeshare to run with Windows 3.0 in protected mode would be available *long before* the drivers for LAN Manager. If we planned to move to Windows 3.0 in the near future, then Officeshare was the choice to make. We did plan to move but not immediately so the next step was to try each. Officeshare was up and running on a PC in about 20 minutes; LAN Manager, on the other, hand took a couple of hours to get to the point where it connected to the HP3000 and then Windows refused to run because there was so little memory left over (this on a 4 Mbyte QS16S). Forget about running NewWave.

So Officeshare it was. With the software finally arriving, we set about our implementation plan. It took a bit of time to actually get the 922 configured properly so that the DTC and the LAN would both operate off the single LANIC, but the HP engineer had the same response centre to call that we had and the network was soon configured.

Another nasty surprise awaited when we tried to get the NS connection to a classic HP3000. We had simply - as we had been assured we could - moved the Bell 201C modem from the old system to the new one and tried to get the network up. However NS on the new HP3000s will not communicate using bi-sync protocol (which is a half duplex) it will only run over dialup lines using LAP-B protocol (which is full duplex). The manual says you can use a Bell 201C modem but what it doesn't tell you is that you cannot use this modem (which is a half duplex standard) on a dialup; it can only be used on a leased line in a full duplex mode. According to internal information from HP (not in the manuals), there are only a couple of modems that will work. HP strongly recommends that the modems be used in conjunction with auto-call units. However NS will only talk to one type of ACU (Bell 801) but the only modems we could find required a different type of ACU! After a great deal of research (and enduring much ridicule from modem vendors at the idea of a full duplex dialup modem), we discovered one combination we believed might work with a corresponding type to be found for another system on the other side of the Atlantic. It took a long time and a good deal of unanticipated expense but the system to system communication now

works very well and we learned a number of things about implementing networking:

- it is a real nuisance to have both DS on classic machines using bi-sync and NS on PA machines using LAP-B on the same network.
- if you don't have NS configured exactly correctly on a classic machine you are likely to have system failures whenever you try to connect to a remote machine.
- if you purchased local support, in our case Net Startup, you are not entitled to call the Response Centre! While the Response Centre would, if pressed, provide answers to specific questions they would not take ownership of problems covered by the local contract. The subsequent delays reminded us of all the reasons HP implemented the Response Centre in the first place.

Back to the main implementation: we had contracted with a dealer to take a truckload of PCs over a specified period and now started to take delivery in small batches to set up to go on users desks. The whole concept began to come more clearly into focus as we started to set up the PCs. With Officeshare the system administrator sits down at a PC and goes through a relatively quick process of making up install discs for each users PC. While the process is a bit tedious and lacks a certain flexibility it is very straightforward (there are a couple of niggling points: you *must load* from drive A, you can't specify the drive you want to use and there is a glaring bug in the procedure in that one critical file is not copied from the master discs to the Network Master you create - you must remember to copy the file LMSET.COM by hand!) The real advantage of the process becomes evident when you start to set up individual PCs: essentially you install Windows, connect the PC up to the LAN and put in the prepared Officeshare install disc. Then you leave it running by itself to load the network software, NewWave and any other components you have purchased as part of NewWave Office (remember NewWave Office is not a single product but rather an umbrella or family of products). Come back an hour later and key in the user name for NewWave on the PC then let it finish up by itself. Nothing could be easier and the process worked flawlessly.

All our proprietary systems from the old series 42 migrated to the 922 without a hint of any problems. They all ran - and continue to run - in compatibility mode with not a single problem. Testimony, I am sure, to the monumental effort HP put into this feature of MPE XL. The network itself has given us no problems

whatsoever: at first we would do a check on each lan card installed in a PC but quickly dispensed with that when we discovered they always worked.

The only time they didn't was when we installed our first Ethertwist card (the original order was for Starlan 10 cards and hubs, we didn't know that by the time they were delivered they would be obsolete) and got a message about a 'problem with the wiring'. The Response Centre knew all about that one without any research and had us going again in a few minutes. Later when we upgraded from DOS 3.3 to DOS 4.01 and the network software would not load ('wrong version of REDIR.EXE'), the Response Centre solved the problem with equal alacrity. A rename - the initial Officeshare installation under DOS 3.3 puts a file called REDIR.401 into the network directory - and we were underway again. Spooling printers on the network is also straightforward and robust, once you discover that any Laserjet after a series IID does not support status checking: if you want to spool a Laserjet III, for example, you must buy a special interface card (HP 26013A Serial I/O PCA) to configure it as a type 22, otherwise it will only work as a type 18 with no status checking and you *don't* want to do this. Last but not least was the performance of the 922 compared to the 42. One big batch job involving a lot of calculations and subsequent database updates runs easily 4 - 5 times as fast as it used to - which was reassuring when considering that performance was one of the basic issues requiring an upgrade.

Now came the shocker: basically NewWave Office didn't work. All of our testing had been with standalone PCs, none had been on a network. The memory problems that were a nuisance for NewWave power users on standalone systems became a nightmare for everybody on the network. That old DOS bugaboo of the importance of the base 640K of memory combined with the bugs in the memory manager of Windows 2.11 were doing us in. What was just manageable without the network drivers loaded was a disaster with the PC on the network. Secretaries doing nothing but Word processing could only have one document open at a time. If they wanted to check their bosses' calendar using a terminal emulator they had to close everything else down first. After being used the terminal emulator had to be shut down before there was a hope of opening up an Ami Pro document. Even at that it was necessary to shut down NewWave and restart it sometimes several times in one day. A few times it was even necessary to switch the AUTOEXEC.BAT file and reboot the PC off the network just to accomplish certain functions. The good news was almost completely lost in the

impact of the bad on the frustrated users (it is horrifying to think what could have happened if the comptability issue had turned out any other way).

Then, to add insult to injury, the NewWave Office 'product' essentially ceased to exist in the middle of our implementation. That is to say Windows 2.11 became unavailable but NewWave Office for Windows 3.0 did not exist (it still doesn't at the time of writing this almost a year after the introduction of Windows 3.0). What was to be a relatively leisurely move to Windows and NewWave 3.0 when we were ready became an obsession. A frantic 'project' to evaluate moving to 3.0 as soon as possible got underway (on evenings and weekends mostly). Word that the Officeshare drivers would be available soon (co-panelist Bill Crow of HP at the Interex conference in Boston stated definitely available by November 30) kept alive hopes of better days ahead, and we pleaded with HP for any news or other help to keep the project on the rails. Meantime we learned that NewWave 2.0 with some (badly needed) Agent capability would never be released which was the last straw. We had to move our users to 3.0. We started out and very quickly but ran into a really baffling problem: we couldn't print to our spooled printers! The Response Centre worked with us for a week on this before we got to the bottom of the problem. The problem turned up over and over again in different guises but is essentially this: upgrading a PC from Windows 2.11 to 3.0 can (and usually does) do terrible, subtle things to the WIN.INI file.

After a lot of frustrating recurrences, in various forms, we learned that whenever you do this upgrade you should either remove (or rename) the WIN.INI before starting the upgrade and then put any special sections back in by hand or be prepared, after the upgrade, to go through the WIN.INI with a fine tooth comb and clean it up by hand. In either case it is necessary to be conversant with the format and content of WIN.INI. A standard procedure for us was to use the Control Panel to remove all printers and then (with NOTEPAD) to go into WIN.INI and manually remove all lines that had to do with printers, then restart Windows, go back to the Control Panel, and reinstall any printers.

Of course we had to run in Real mode because we were using NewWave mail and using Cooperative Services for the systems we were developing. Real mode still has the same old memory problems albeit they did not seem as severe as 2.11. And Real mode is slow. Real mode is **SLOW.**. Especially if your SMARTDRIVE parameters are not quite right. Opening up objects could take 20 -30 seconds,

closing an object almost 10 seconds! Considering the amount of opening and closing required because of the memory problems things were going from bad to worse! Another problem arose with printing: it could take 15 minutes to print a 12-14 page document! So much for productivity gains. The abstract for this paper printed on a Laserjet series II from a QS16 took over seven minutes to print when NewWave was in Real mode! We learned, through a lot of tedious benchmarking, that in enhanced mode with the latest Laserjet III driver (compliments of - who else - the Response Centre) printing could be up to eight times as fast (about 50 seconds for the above mentioned abstract). Opening and closing objects was 2 - 3 times as fast in enhanced mode and closing objects at least 3 times as fast. And you needed to worry less about constantly opening and closing as you could have a dozen or more objects open at once, fewer if on the network but still more than is reasonable on a normal VGA screen (large screens have got to be in my future). By now our pleading with HP was turning into screaming for a solution but no response was forthcoming except that someone had been asked to "call the Division".

Then in January the blow fell. Officeshare drivers for 3.0 would never be released! Just what were we supposed to do about networking, the terminal emulator (Advancelink), NewWave mail and about applications ready to implement using Cooperative Services? The answers came slowly; the basic networking, it turns out, will work whether you are in Real or enhanced mode. There is one catch here though if you use DOS applications: to run in enhanced mode you do not use any memory driver (except HIMEM.SYS) so that Windows can discover and use for itself all the wonderful megabytes of extended memory available. This means that when you load Officeshare there is no EMS memory so it loads into regular memory leaving that much less for any DOS applications. For NewWave Mail we would just have to live with having a limited number of users reboot their PCs to run in Real mode process incoming and send outgoing mail before rebooting to go back to enhanced mode. A new version will be available running over a LAN Manager network perhaps as early as April (at the time of writing the rumour is that it will be May or June - fully a year after the introduction of Windows 3.0).

As for Cooperative Services no one at HP, even the few people who had heard about it and knew it was an HP product, had any idea of what would become of it. Meanwhile we had a critical application developed and ready to implement based

on Cooperative Services! We could not wait however many months it would take for LAN Manager to appear and then rework the application. The same story applied to Advancelink (Session): there was just no indication from HP as to if and when it might work in enhanced mode. We now fully understood we had to find our own solutions to problems wherever we could. A mad scramble ensued to get beta versions of PPL (and PPL Toolkit) to replace Cooperative Services and R1WIN as a terminal emulator from Walker, Richer & Quinn. Thence to convert the application to run over the LAN using PPL. To this day when you order a NewWave developers kit it comes with a manual called **HP NewWave Environment General Information Manual** that spends a good deal of time talking about Cooperative Services as the way to integrate your PC applications with the HP3000!

Then came the process of converting users to enhanced mode (and coincidentally upgrading to DOS 4.01). Another lesson learned, mostly the easy way, was the importance of standardizing as much as possible your PC hardware configuration. Every configuration has it's snags and quirks, and, the farther you get from average, the more problems you'll find. We (and the Response Centre) spent a month on a particular problem with Lotus in NewWave on a 20" IGC monitor. We were the ones that finally tripped over the solution (by this time the Response Centre was busy trying to assemble the exact same - somewhat unusual - hardware configuration we had) but we wouldn't have known it was the solution without the information we had already received from the Response Centre.

Unusual or non-standard configurations can consume enormous amounts of time and energy whereas if all your PCs are identical once you have figured out one you've figured them all. We've also learned 386 machines are much easier to work with than 286 machines. If you are 'standardized' on 286 PCs don't implement anything that relies on Windows 3.0. Moreover, I suspect that if you are working with Vectras rather than brand X PCs, you'll get farther faster with the HP response centre, particularly in the murky area between hardware and software.

Did we accomplish what we set out to do? Did we realize our vision of users interacting with the system in a single intuitive and seamless way? Essentially the answer is yes although we are by no means finished. During implementation the

day suddenly came when, working through a list of user complaints, the 'problems' were all user problems (a paper tray resting on the shift key made selecting text on the screen a cumulative thing, for example).

New users are now implemented in a routine fashion, and the problems that do crop up are usually procedural or ones we've seen before and know how to deal with.

Now we can get down to the business of seeing what'll she really do. There really are substantial productivity gains to be had. Exactly what will be valuable to any particular organization will depend on which application software is being used and the needs of the organization. In this implementation the basic networking functionality provided our first big boost:

- we set up common shared object storage for work groups (yes this was available in NewWave 1.0) that eliminated the Adidas network for sharing Lotus spreadsheets. Executives started using the word processing to type drafts of their own memos then drop them into the object storage shared with their secretary.

We started educating users to use the functionality the environment gives them

- users started adjusting their screen colours.
- users started to keep more than one object open at a time.
- users organize their objects by colour of file folder.
- we discovered the convenience of being able to drag several objects at once to the printer just before leaving for lunch, a meeting etc.
- we started using simple Agent tasks for repetitive things and for convenience such as the Agent task supplied by HP (in the What's New folder that comes with NewWave 3.0) to schedule reminders. We had a problem with an external mail system that was coped with by having the Agent, every morning Monday to Friday at 10 a.m. rain, shine or sickness of the owner of the PC, mail a message off to a remote user.

Gradually the more sophisticated users, and the ones with the more sophisticated needs, began to explore the possibilities inherent in NewWave. For example:

- a long memo (in Ami Pro) required an organizational chart that had been done in Graphics Gallery: nothing could be easier in NewWave. You can simply drop it into a frame, copy and paste or share.
- It is simple to demonstrate each of these facilities and then undo it to demonstrate the next thing.
- the CBT, so valuable getting people started in NewWave, was used to create tutorials for the applications developed.
- more complex Agent tasks were developed to assist in day to day work such as rummaging through the file drawer, opening up folders, looking for any object whose title contained a word or words supplied by the user.
- Agent tasks are used to process incoming E-mail from external systems.

We are convinced that object oriented systems, with all that they imply, are the way of the future. It's a rule of thumb that the simpler a system is made to appear from the user's point of view, the more complicated it is behind the scenes and this is certainly true of NewWave Office. There is a long way to go to fully realize all the possibilities but the technology as it exists today is very rich and is certainly worth implementing. Successful implementation today means being poised to take advantage of new developments that appear tomorrow. In the meantime I don't think you could take our users PCs away from them and replace them with Macs.

DISTRIBUTED PROCESSING AT HARVARD UNIVERSITY

Paper #4109

Joel Martin
Harvard University
Development Computing Services
124 Mount Auburn Street
Cambridge, Massachusetts 02138

Tel: (617) 495-0455

Abstract

Distributed applications and databases are getting more attention now that personal computers and network servers have become more capable. This paper will share the lessons learned in the Harvard University Development Office about distributing applications in a mixed environment. (Our office has three HP-3000s and 180+ personal computers. We are also developing linkages to an IBM 3090 and six or more graduate school microcomputer LANs.) The topics covered include: advantages and disadvantages of distributing your processing, how you may have already distributed (whether you know it or not), and the steps that we followed in developing a distributed approach. Special attention is given to our redefining the role of the HP-3000 and the impact of that on our purchases of hardware and software, leveraging the advantages of each platform and using currently available products to start distributing today.

Background

For most of its 350+ year history Harvard has had a strong tradition of decentralized decision making. Each of the graduate schools operates autonomously from Harvard College, managing their own income, expenses and therefore their own computing facilities in the style of independent corporate subsidiaries. This has led to the growth of dozens of autonomous "islands" of computing with computers from a very wide range of vendors.

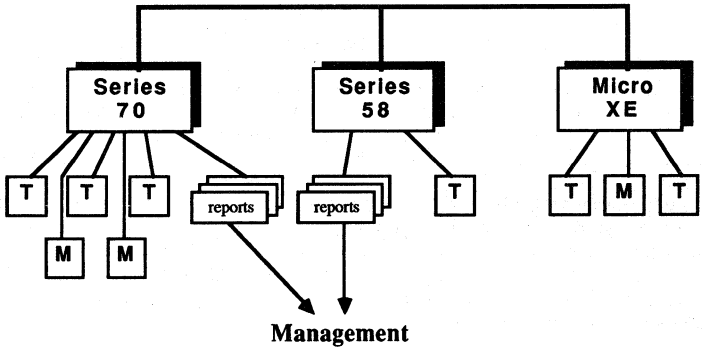
In 1987 one such island was the University Development Office with an HP-3000 Series 68 that was host to the address, biographical and gift information on all of the University's more than 300,000 alumni and friends. In the spring of 1987 we brought the major portion of our primary new application on-line and found that the Series 68 could not support both the on-line and batch functionality. (The application is about one million lines of code and required seven gigabytes of disc storage.) After a series of meetings with our vendor, Hewlett-Packard and independent consultants we proposed to senior management an upgrade of the Series 68 to a Series 70 and the purchase of a Series 58 and a Micro 3000 XE. (At that time the 900 series computers were not yet shipping and the other alternatives we explored were not feasible.) With the two additional minicomputers and the ongoing installation of microcomputers we were able to offload work from the Series 70 in three ways: 1) all word processing and single user functions were moved to the microcomputers, 2) all software development and testing was offloaded to the Micro 3000 XE and 3) a significant amount of our batch reporting was moved to the Series 58.

The only novel element in the new scheme was our use of the Series 58. We established a complete mirror image of the Series 70 databases on the Series 58 and purchased the Backchat software package (Proactive Systems, Ltd.) to bring changes from the Series

70's databases to the copies resident on the Series 58 in real time. Our Series 58 was then used to run batch reports exclusively. (See Figure 1.) In addition to freeing the Series 70 for our on-line users we discovered a number of side-effect advantages to a multiple CPU implementation, including very high availability for reporting and simplified performance management. (See Table 1 in the tables section at the paper's end.)

University Development Office Network

(First Distributed Stage)



Series 70	- On-line transactions, 50 - 70 users
Series 58	- Large batch jobs, 4 - 6 users
Micro 3000 XE	- Software development

(T = terminal, M = microcomputer)

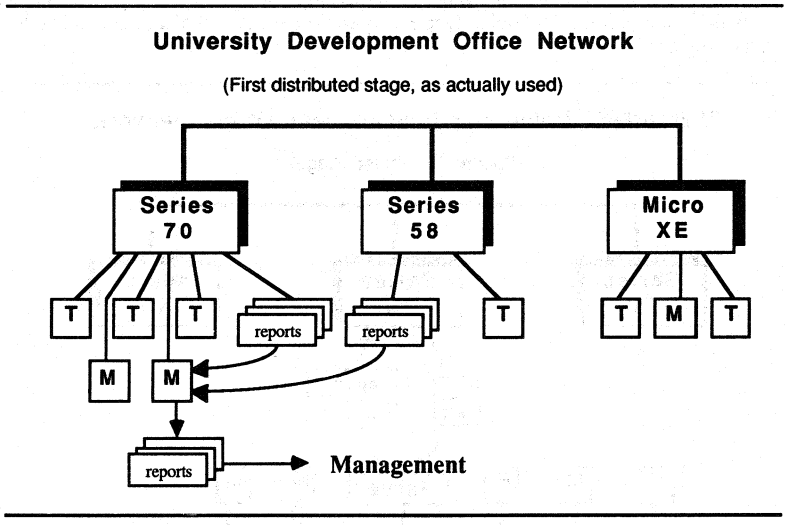
- Figure 1 -

Once our computer installations were completed, we thought we would be free to focus on the as yet unimplemented parts of our HP-3000 based system, on offering on-line access to our system to the graduate schools and on "linking" our system to the other mainframe and minicomputers on campus. But a revolution in computing had taken place between the time of our minicomputer applications design in the early 1980s and our readiness to offer access to the graduate schools. Our potential external users were no longer willing to settle for the standard on-line screens and reports, no matter how many options for selecting, sorting or formatting we offered. They were interested in owning a slice of our data that they could manage locally on a personal computer or microcomputer network. We were being pushed by all of our users and potential users to download data for them, though we had not yet addressed concerns about data security, incorrect interpretation of the data elements, the meaning of data element values and the inevitable data reconciliation problems that would arise. Some managers were fearful that if users had local control of the data to produce their own reports then we would be creating our very own Tower of Bable problem, should two or more of these users bring reports from their own systems to the same meeting.

Opportunities and Problems

As we wrestled with the dilemma of downloading versus retaining control of the printed figures, two events took place that pushed us further down the path we were to follow. First and foremost, we had external users at one of the smaller graduate schools approach us about installing a microcomputer network based alumni system that they had determined would meet their needs if we could perform an initial data load and work with them to manage the synchronization and reconciliation of any differences that might arise. (This school was tracking its 20,000 alumni and their gifts on index cards. Address changes usually made it from their cards to our computer system via a preprinted paper form. Staff from this school did review our minicomputer based system prior to reviewing the microcomputer based system, but they found the functionality, flexibility and local control offered by the microcomputer based system to better fit their requirements.)

The second event that took place was a study we undertook that highlighted how information was being used by our local users. (Our "local users" are those with direct on-line access to the application running on the HP-3000.) We found that many staff were working on a microcomputer with a subset of the HP-3000 information in a manner very similar to the smallest graduate schools. (See Figure 2.) This was done in spite of having access to the screens and reports on the HP-3000 that were written to support their needs.



- Figure 2 -

The only significant difference between our local and remote users was that our local users were using spreadsheet files on microcomputers instead of "index cards", microcomputer based servers or separate minicomputer systems. The mechanism for updating their data was the same as that of our remote users - information from reports used for processing and auditing transactions was used as the source data for changing spreadsheets. The spreadsheets were then used to create executive level summary information. Our own users had "distributed" to another level without any assistance from us! (Don't laugh too soon - your users probably have, too. See Table 2.) In the most extreme case a nightly

report from the Series 70 was used for data entry into a simple, user specific TurboIMAGE database that also resided on the Series 70! (A virtual server!) When that user was given a microcomputer the database was moved from the Series 70 to the microcomputer, but the user continued to retype information from the report into the new host. (In fact, the user had no choice but to retype the information if the specific reports requested by the user's management were to be produced.)

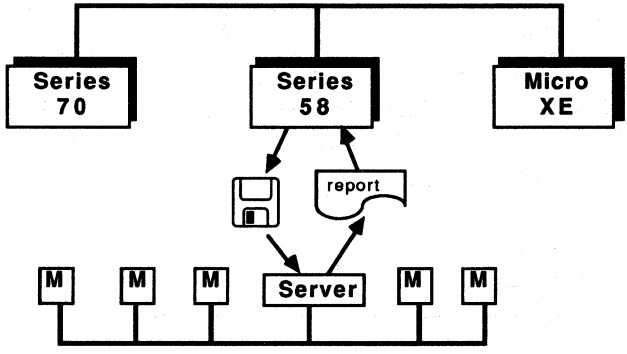
Reassessment and Enlightenment

When this information came to light we did some rethinking of the role of our minicomputers and our own microcomputers in filling the needs of our internal and external users. Our internal users had really created their own "servers" by retyping data from their printed reports into their desktop systems. The similarities between our external user's microcomputer server, our internal users desktop "databases" and our mirroring of data onto the Series 58 started to become apparent. In many ways the microcomputer server was a natural extension of the distributed view we had embraced with the installation of the Series 58.

With a fresh perspective we approached the linking of our pioneering user's microcomputer system to our HP-3000 system as a prototype for serving our non-data entry internal users as well as the other graduate schools. Our configuration at the end of our second stage is reflected in Figure 3.

"Extended" University Development Office Network

(Second Distributed Stage)



- Figure 3 -

The largest hurdle to downloading to a server was how to reconcile data between the two systems. This proved to be less of an issue than management had feared. (Especially after it became apparent that we already had a number of staff creating their own reports from

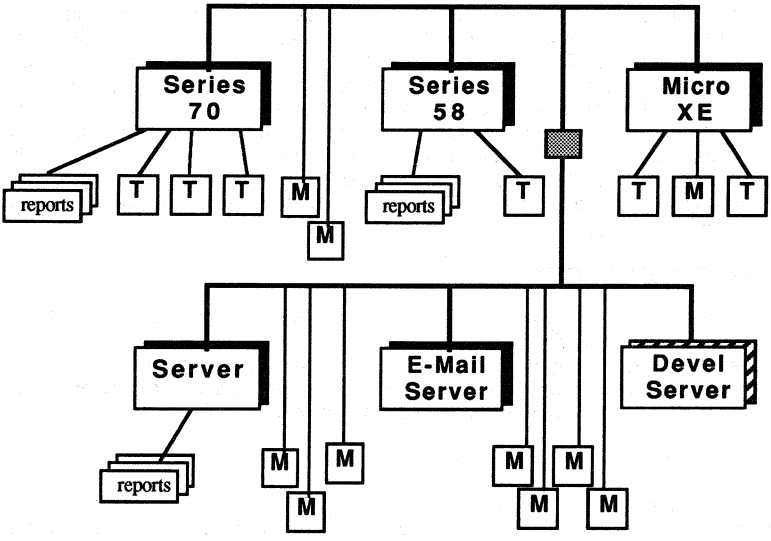
data they had retyped.) A procedure was defined for reviewing and resolving discrepancies. The procedure largely consisted of identifying the contact person in each office and attaching a change origination code and date to each change. The contact people would then determine the cause of the discrepancy and would change either the server's view or the HP-3000's view accordingly. On a quarterly basis we would unload data from the HP-3000 in order to perform an audit run against the data on the server. (It would later turn out that the quarterly computer resynchronization would not be done. The daily and weekly mechanisms for identifying discrepancies were proving sufficient, and some degree of error was acceptable and expected due to the frequency with which alumni change addresses.)

A Server of Our Own

The advantages to moving real information to microcomputer based platforms (in addition to the office automation functions) became too persuasive to ignore. (See Table 4.) Though our external user did not have an information systems staff he did have a more sophisticated information system for his use than did our in-house users. (This is certainly not to suggest that the utility of a system is in inverse proportion to the number of information systems staff supporting it.)

"Extended" University Development Office Network

(Third Stage)



- Figure 4 -

During the past year we separated the managerial information portion of our alumni system from its transaction processing base. The managerial system runs on a Unix workstation server. The server network is attached to the HP-3000 network by an HP LAN bridge. Users of the server based system can also access the screens and reports of the HP-3000 based system by logging on over the local area network. Note in Figure 4 how similar our microcomputer network has become to our HP-3000 based network.

Lessons We Learned That May Help You

Many of the following admonitions are familiar ones that may well be incorporated into all of your project plans. They are all the more important if you are making the jump to a mission critical personal computer based system. In a general sense all of these items fall under the category of reducing your risk.

1) Follow a path that is in step with your corporate style and strategy.

   The key to our approach has been the decentralized nature of Harvard. Though there are two "corporate" IBM mainframes, each graduate school ("cost center"?) has flexibility and budget enough to pursue their own course and at Harvard that usually means a path that allows local control.

2) Locate a user willing to be the pioneer that has time and money available.

   Or better yet, have a user locate you! Being the pioneer means suffering some hardship and tolerating the inevitable problems that arise in crossing a new frontier. The user must be absolutely committed to the project's success, patient and willing to accept the first system as less than ideal. This commitment has to extend to devoting user staff time to ongoing support after the system is online. We are still using floppy discs as the transport medium to and from the microcomputer system (though that is expected to change with the installation of a University wide network), but our inability thus far to "complete" the interface has not reflected badly on our department or the project for two reasons: 1) the system works and meets all user requirements including those of timeliness, and 2) the user knows that to cause a fuss about the interface would reflect poorly on the project and thereby reflect poorly on his department.

3) Keep the first project simple.

   Simple does not have to mean small in contribution. Simple does mean that you have taken steps to reduce the risk of real or perceived failure. Simple also means planning to have a well-defined and well-bounded set of functions operating within a year at the most. And simple means you have not (yet) committed to a huge capital expense. (If necessary, artificially limit the number of users in your first project.)

4) Buy it, don't build it.

   And if you can't buy it, then use the highest level language the application can support. We used SUPRTOOL and Quiz for the minicomputer export programs. The minicomputer import programming was done in FoxBase. Files will be "bridged" from the minicomputer to the microcomputer server using a commercial terminal emulator.

5) Put your best staff on the project.

Your first experience will mean some level of trial and error and will require more imagination and innovation than the "typical" project. These staff members should be free of other responsibilities if this is your first experience with a new technology or type of application. This is just as important a guideline whether you purchase a package or are forced to develop the application yourself. The staff needs to be led by someone knowledgeable about the business area who is credible to the user, energetic, an effective communicator and careful about setting user expectations.

6) Follow a methodology and define the existing work and information flows.

This might be an uncomfortable experience, even for your highly committed user, if the current system is in an embarrassing state of disrepair. (There is a high reliance on scraps of paper and other informal systems, there are few auditable points in the process, etc.) It is very important that all the details are documented or your new system may never get fully off the ground. Of course documenting all of the details does not mean that all steps in the process have equal weight or priority. You may find a number of areas that could be improved that are better left undisturbed, at least in the first implementation. Change can be frightening, and pushing for functions that are not user priorities could put your user's commitment to the project at risk. Your methodology must include definition of the project's objective by the user in business terms. If the user is unwilling to define the objective on paper then you should have serious reservations about the level of user commitment. The user should also commit staff to assist in the testing of the software, the writing of the documentation and the training of other staff. The degree of success you can expect is directly related to the level of active user participation in all phases of the project.

7) Focus on the core data.

You are distributing because the informational and data needs between or among the groups are different. Define clearly the key areas of data overlap, and get your users to commit to one definition and one control point for each of these items. One thing to keep in mind -- even within your core data there may be wide variances in the frequency with which the data is updated or needs to be resynchronized. On our HP systems Backchat keeps the Series 58 current with the Series 70 to within a few minutes unless we run a huge batch update on the Series 70. For the microcomputer system we are content with weekly or monthly synchronization. Core data may be defined as all values of some data items (vertically distributed), some values of all data items (horizontally distributed) or a combination of both.

Part of the core data issue is defining the appropriate level of data. If your microcomputer system is going to serve the needs of upper management, then you may not need to store all of the detail for any view on your microcomputer system. To get a sense of what is necessary review the reports that management uses and the report changes that they have requested. You may be able to make do with summary level information in many areas. (An advantage to limiting the data you bring to the microcomputer is improved performance.)

8) Provide contingencies and controls in the data transfer.

If your computers are linked via a wide area network, then wonderful! But be sure that if the network goes down you do not need to perform a complete resynchronization of your core data. We are still largely using magnetic tape and

floppy discs as the medium for our transfers, but it works. Remember to consider all of the minicomputer (glass house) issues in defining the linkages and operation including security, back-up, tape or disc storage, restart and recovery procedures, etc. As a start, make sure that the server is kept in a locked area accessible only to authorized staff, and be sure to have a maintenance plan in place well before you go on-line. (In our first distributed project we had two server disc drives die in the first month. If this is an important application you should be as careful and rigorous in your support planning and back-ups as you are for your minicomputers and mainframes.)

9) Take advantage of each platform's strengths, and work around each platform's weaknesses.

Our microcomputer application is much easier to modify than our TurboIMAGE and Cobol minicomputer application, so much of the programming logic in our data transfers is handled on the microcomputer end. The HP-3000 is much better suited to large print jobs than the microcomputer, so we still generate mailing labels for the graduate school when they want to do a mailing to all 20,000 alumni. We had two disc drive failures on the microcomputer server in the first month of operation, versus one in four years on the HP-3000. (And we have 14 discs on our HP-3000 versus only one on the server!) Even though we had taken several precautions in arranging support for the server hardware there were still delays in getting it repaired.

10) Automate and document the whole process.

You will need to define a process that includes updating, auditing and resynchronization of the data in addition to providing the initial data load. Put everything you can into jobstreams and macros. Document the steps to be followed with checklists for your operations staff and users. Have your user document all procedures to be followed in their area. Get a commitment on user development of the documentation at the outset of the project, and take care that the documentation is developed concurrently with the application software. This is not to suggest that you abandon the documentation to the user, but you want to make certain that they feel ownership of the documentation and thus of the system. Having the user develop the documentation should also instill in them a greater respect for your efforts in developing the system and a better appreciation for the complexity of your future software development undertakings.

Finally, add your own items or goals to this list and keep it at hand to help guide your decisions when the going gets rough.

Meanwhile, in the outside world...

Our approach to distributing information is by no means the only one open to HP-3000 users, and (as one might guess) is not even the only path being pursued at Harvard. The Financial Office has developed a very effective and well received microcomputer interface to data residing on the IBM mainframe. One strong similarity between that system and ours is that a separate database was created to support management financial reporting and the transaction processing system was left largely untouched. This "financial workstation" uses an off-the-shelf front end and communications package to retrieve data using SQL from a DB2 database. Data retrieved through this method can be printed on the mainframe laser printer, viewed on the microcomputer, printed on a local printer or saved as a file for

importation into a spreadsheet or local database. We are looking into the possibility of making data from our HP-3000 accessible to authorized users via the mainframe. The mainframe would in effect become a server subordinate to our HP-3000 Series 70. Data would be passed between the computers in a manner similar to the Series 58 and the microcomputer servers.

One of the graduate schools at Harvard has proposed developing a system for their needs that would use an HP-3000 Series 850 (the Unix version of the Series 950) as the host with new HP Series 700 workstations, also running Unix, serving as the clients.

Yet another approach we are exploring is to provide a tool of our own like the financial workstation that would directly access data on the HP-3000. In our case the host computer would likely be the Series 58 or its MPE-XL successor. We have had good success with adding indices to the mirrored databases on the Series 58 to speed the production of some reports without afffecting speed of transaction processing, which takes place on the Series 70.

Good Luck

I hope our experiences will give you some benefit and a measure of courage in tackling your own distributed processing needs. Do what is right for your company and your department, and don't get caught up in the rush to this year's most popular panacea if it does not fit your needs.

Advantages of Our Multi HP-3000 Implementation

• Capacity	The three minicomputer configuration gave us the capacity to handle the existing load and the growth in data and applications forecast for the following four to six years. We may reach the capacity ceiling next year, five years after the two new HP-3000s were purchased.
• Increased Functionality	The shift of word processing from HP Word running on the HP-3000 to Microsoft Word brought a quantum leap in functionality and ease of use. (Certainly no surprises here, but this was news to our users four years ago.) Other functions, e.g. graphics, saw even greater benefits.
• Response Time	Everyone's response time improved as a result of splitting functions onto separate computers. Furthermore, the response times in each area (word processing, batch jobs, software development, etc.) were now independent of the work load in each of the other areas.
• Availability	The Series 58 is available for batch reporting for 23+ hours each day because it's databases, being duplicates, do not need to be backed up. Should our situation require on-line access of 23 hours or more each day then we can accommodate that by backing up the Series 58's databases instead of the Series 70's.
• Upgradability	Each CPU is separately upgradable without impacting the other CPUs. We will be able to migrate our software development (Micro 3000 XE) to a Series 900 and shake out any problems before migrating our production systems.
• Variability	With separate CPUs for the major functions each CPU is "tuned" to perform that function best. At first this just meant the changing of job limits and job queues. In the spring of 1991 it meant some limited restructuring of the Series 58 databases with OMNIDEX. In the future it could mean running AllBase, SyBase or Oracle on the Series 58's successor while still mirroring changes from the TurboIMAGE database on the Series 70.

- Table 1 -

No table such as this would be complete without a counter-listing of the disadvantages. The disadvantages were fewer (and considered less important than the advantages), but in another environment they may have been weighted differently. See Table 2.

Disadvantages of Our Multi HP-3000 Implementation

• Cost (?)	There was a significant cost associated with the move to microcomputers and two more HP-3000 minis. Beyond the cost of the hardware there was the investment in additional copies of software packages and tools. If a larger HP-3000 had been available it is still unclear whether the cost of upgrading would have been significantly less than distributing or if the advantages of distributing would still have outweighed the savings.
• Operational Complexity	Adding two HP-3000s meant an increase in the complexity of our computer operations, but this proved to be almost unnoticeable. Our operation had already been largely automated and the steps necessary to synchronize the mirrored databases were easy to follow though large in number (due primarily to our application's database complexity). We did not hire additional operations staff.
• Perception Change	This was in some ways the largest disadvantage for us. With three CPUs we were now viewed by some as a large, perhaps unresponsive, mainframe operation.

- Table 2 -

Have Your Users "Distributed" Without You?

- Do far more users have spreadsheets on their microcomputers than work with budgetary or accounting information?
- Do your users enter data from MIS reports into their own spreadsheets because the desired "user view" is not available?
- Do you have an applications software backlog? Have your users stopped expressing frustration with your backlog?
- Has the ratio of administrative staff to managerial staff increased?
- Are your administrative staff more computer knowledgeable than your managerial staff or their predecessors?
- Do any of your users have database software (Dbase, etc.) on their microcomputers?
- Do your users keep old reports and listings in their desk drawers? Do they update these reports by hand?
- Do you have software change requests that seem intended to make reports more workable as data entry documents?
- Are you printing more copies of detailed reports than would seem necessary for processing transactions or auditing?

- Table 3 -

Mainframe/mini or Microcomputer Server?

	Traditional Mainframe or Minicomputer	Microcomputer-based System
Support	Good to excellent.	A potential problem. You may have different vendors for the server, the LAN, the network operating system and the database management system. Ouch!
Ease of use	Excels at repetitive functions that will change little over time (e.g. data entry).	Excels at adhoc inquiries, information manipulation and presentation (scrolling windows, on screen graphics, color!).
Ease of modification	Poor (COBOL w/ IMAGE) to good (4 GL or relational)	Good to excellent
Capacity	100s of users, gigabytes of data	10-30 users, 100s of megabytes of data. The gap is closing in most capacity areas, except for volume printing.
Databases	Hierarchical or very expensive, but well suited to high volumes of similar transactions.	Relational and inexpensive - well suited to adhoc inquiries, presentation quality reports.
Word processing	Poor to fair	Excellent - powerful, sharable, easy mail-merges, beautiful output, WYSIWYG.
Security facilities	Excellent, esp. with third party additions.	Fair to very good and improving.
Reliability, restart and recovery	Excellent - very high MTBF, excellent power fail restart, instrinsic level safeguards, transaction logging, HP support.	Fair to good, but less of a potential problem - much lower transaction volumes, fewer users, our system is not "University critical"
Supporting applications (word processing, spreadsheets, etc.)	Poor to fair. Tools are certainly available to do large mail-merges from IMAGE to your local Laserjet, but is it worth the effort?	Excellent. The newer, better and easier to use end-user applications are appearing on the microcomputers first, and integrating information from disparate sources is much easier.
Strengths	Processing high volume of similar transactions.	Tools for user reporting, versatility and reconfigurability, presentation quality output, price/performance.
Best-suited applications	High volume, relatively static, interdepartmental, mission critical, frequent and regular use by operational staff.	Dynamic, departmental, information or knowledge enhancing, adhoc/irregular use by managerial staff.

- Table 4 -

TITLE: EDI Diskette Transfer - The Connectivity Issue

AUTHOR: Ken Nutsford

Timeshare Systems Limited

P.O. Box 60032

Auckland, 1230

NEW ZEALAND Phone +64 9 799 180
Fax +64 9 799 180

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 4110

Hook me up, Scotty - Towards the Enterprise Network

Anthony C. Furnivall
SDL/Software, Inc
P.O. Box 601
BUFFALO NY 14222

INTRODUCTION

Information management, like other fields of human endeavour, is fond of categorizing periods of time to describe the trends which became apparent during that period. Thus we have the sixties as the era of mainframe computing, the seventies as the era of departmental computing and the eighties as the decade of personal computing. A certain caution should perhaps be evident in trying to apply a label to a decade which is only one or two years old, depending on how you read your calendar. However, it would not be too extreme, I think, to label the nineties as the decade of networked computing.

The trend has been visible for a long time, with elementary data communications techniques becoming more and more refined, and a new network infrastructure being developed to tie together disparate computing resources. However, the important phenomenon which will emerge in the nineties, is the rise of a network which ties together computing needs and capabilities for *all* employees of a business enterprise, and does this with no technical demands upon the end users.

This paper is an attempt to help define an enterprise network, and to describe some of the problems and issues involved in implementing and managing such a beast.

DEFINING AN ENTERPRISE NETWORK

Perhaps the only other phenomenon which has as many different definitions as enterprise networking is object-oriented programming. If we tap industry leaders and ask them their definition of enterprise computing, we get almost as many different definitions as respondents. This effort was actually undertaken in a recent article.[1] Here are some of the responses:

- *An application process that executes in part on an intelligent workstation, and in part on other processors.*
- *An environment where all information services are available to any user, on any desktop, at any location.*

---

[1] "Co-operative processing and the enterprise network" John Gantz - Networking Management January 1991

- *Transparent access across an enterprise consisting of multiple networks, machines and software applications such that the computing resources of the enterprise become one system.*
- *Transparent access to network resources, development of distributed network applications, and the facilities to create and administer multi-vendor networks.*
- *A style of computing where all of the resources on the network are available to all users as though they were local to the user's workstation*

Each response contains a kernel of truth, but as with most matters of opinion, there is no substitute for coming up with your own definition. There are some common threads, however, and among these we can discern the following:

- **An inclusive approach towards network resources.**
  It is more likely that a resource will be considered a part of the enterprise network than that it will be excluded.
- **A distributed approach towards computing.**
  It is more likely that several CPUs will participate in the total computational effort for solving a problem, than that a single CPU will tackle the problem alone.
- **A multi-vendor approach towards methods and machines.**
  It is more likely that more than one nameplate will appear on the components of the network than that a single vendor will supply all the bits and pieces.

How do we define an enterprise network

All networks are designed to do the same thing - to bring things together. It may help, however, to examine some functional definitions of a network, and see what they bring to our attempts to define an enterprise network.

- **Networks connect resources together.**
  At a personal level, the networking that takes place at a conference such as the INTEREX conference brings *people* together, and allows them to learn more than they would in isolation. At a computing level, networks tie *machines* together, and allow them to achieve more than they would in a stand-alone environment.
- **Networks combine capabilities.**
  At the simplest level, the sort of symbiosis that we have just discussed is a way of combining capabilities. But it is also possible to combine capabilities in other ways. A common example is the use of a protocol

router to allow Macintosh computers on an Ethernet network to access a LaserWriter printer on an Appletalk network.
Another example is the way in which many client/server applications operate - with different machines providing different services to a wide range of clients.

Networks control interactions.
Network security is currently a hot topic, largely because it is so little understood. Compared with the security provisions of almost any minicomputer, network security facilities are limited. But even within these limited capabilities, the network operating systems provide a fair degree of control over the actions of people who are using the network.

All of these functional definitions are common, and frequently used. When we come to apply them to a definition of an enterprise network, it helps if we find a business analog for each of the terms. Thus, while simple networks combine people, enterprise networks combine business entities. While simple networks enhance personal productivity, enterprise networks enhance general business productivity.

From a functional point of view, enterprise networks are largely about integrating people into business activities.

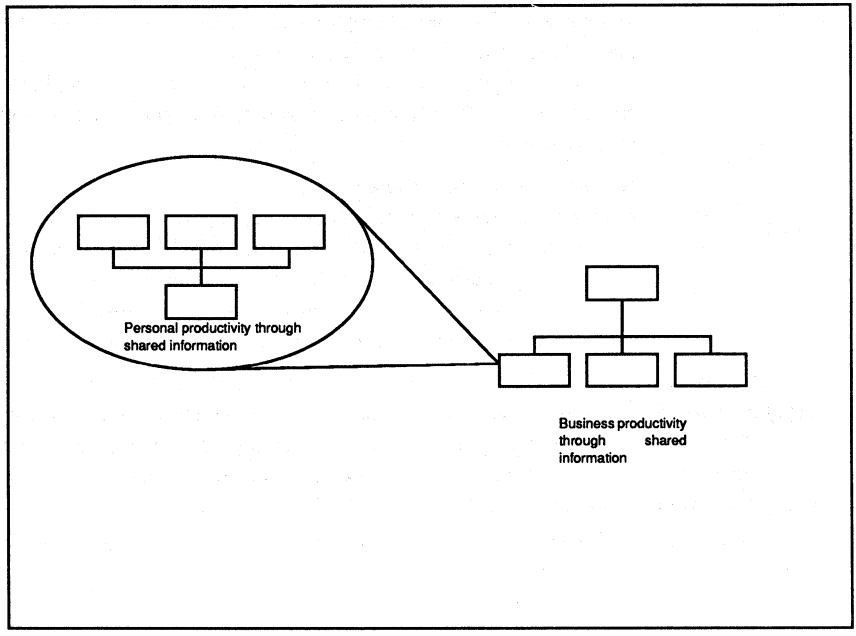


Fig. 1 - Enterprise networking in perspective

A working definition

When we come to take concrete steps towards defining an enterprise network, it will help to consider three entities. These entities exist quite apart from any physical network equipment, hardware or software, but together they constitute a useful way of structuring progress towards an enterprise network. The entities are:

- A sound network plan
- A clear conceptual model of the network
- A set of standards and specifications

What I'd like to do is to consider each of these in turn. Each of them is important, for intuitively obvious reasons, but sometimes the intuitively obvious gets overlooked.

A plan is important because the network is not going to spring into place overnight. There will be times – a lot of times – when the network is only partly in place. In order to retain a flexibility of approach, it is important that you always know what your current goals are, and this requires a plan.

A model of the network, and an awareness of the function of each part of the model is useful, because it becomes possible to place things into perspective. When senior

management raises questions and concerns about your progress, it is easier to relate those questions to a model, when preparing a response.

Standards and specifications are important because they remove one level of uncertainty from the planning and management process. Standards are sometimes regarded as an inhibition to creative expression. This is not the case – they remove from the creative mind the need to cope with drudgery, and free it up for more worth while endeavours.

IMPLEMENTING AN ENTERPRISE NETWORK

The network plan

The network plan should contain a concise statement of what will be achieved as part of the implementation. This statement should be based on four foundations:

- Know your business needs
- Know your mission critical network applications
- Know what you already have in place (if anything)
- Know where you can get good training

Planning is frequently the place where technically oriented people begin to squirm in discomfort. It is not always a painful process. The first two parts of the plan can be created simply by talking to people in your company. Develop a sense of where your company will be in three to five years, because it may take that long to fully complete the process.

Find out what absolutely must happen in order to keep the company alive. Despite the best efforts of controllers, this is frequently *not* the process of taking in cash! In most environments, the ability to deliver your product is the over-riding business need. That product may be software, garden fertilizer or class-room instruction. However, if that product can not be delivered, then you will have no customers to pay those bills!

A secondary benefit of identifying the critical applications is that you identify two key areas. Sometimes it is possible to score a quick, highly visible success, with little or no effort. 'Magic' tricks such as these can enhance your credibility, and increase the support of upper management. The other benefit of identifying your mission critical applications is that you know the areas where special care will be needed. It may be a good idea to delay implementation in these areas until you have some experience, or some success to lean on.

It is vitally important to know what network facilities you already have, and what is the attitude of senior management to possibly replacing it (with whatever your plan calls for). If your plan calls for an Ethernet backbone, and your company has invested several hundred thousand dollars in token-ring facilities, you had better have a good reason for planning the change! On the other hand, the cost of setting up a small network to experiment with different operating systems, or other network components, is an investment that could repay itself several times over, in the years ahead.

Lastly, identify a source of network training. No-one can be expected to know it all, and the best source of up-to-date knowledge is frequently a training class. These classes are not inexpensive – training has become a high-dollar business. However, the intangible benefits of having a week with colleagues in the same situation as yourself is sometimes more valuable than the direct knowledge that you will receive.

A three layer network model

I'd like to present a simple model which has helped guide the development of enterprise networks. This model takes as its basis the familiar ISO model, with its seven layers (Fig. 2). Using this base it defines three layers to help place the issues associated with implementing an enterprise network, into context. The three layers are:

- The Base layer
- The Operation layer

and

- The Enterprise layer

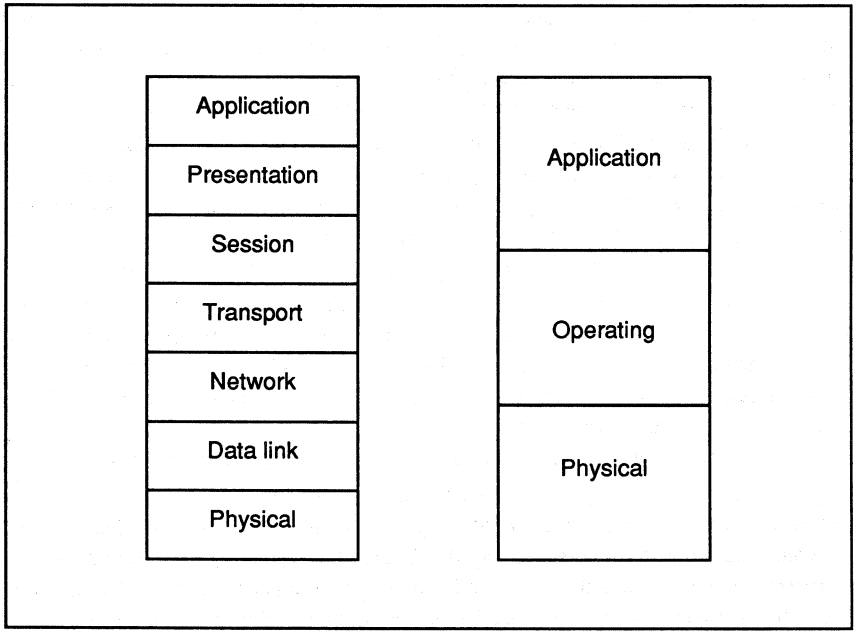


Fig. 2. Enterprise network model

Base layer - infrastructure, wiring and network connections

The base layer helps define the way in which different networks, and network components will hook in to the enterprise network. It includes such things as deciding whether or not you will centralise the placement of servers or how you intend to accommodate bridging between different protocols or transmission methods. The most common transmission protocols in use in HP sites today are probably the following:

- Ethernet/IEEE802.3
- Token ring
- Local talk

There is an abundance of boxes for interconnecting these different protocols, and when you decide on how you will be running your enterprise network, you will need to give thought to where, and how such interconnections will take place.

Another concern which gets addressed in this layer is the concern of portability. People move from one part of a building to another, or they change from one type of work-station to another. Some people are even fortunate to have two different types of work-station available to them! If portability is an important aspect of your network, you will need to come up with a wiring scheme that can handle this portability. In much the same way that you can plug an electric hair-dryer into any electrical outlet, you should be able to plug a work-station into any network outlet, without making extensive reconfiguration changes to either the network or the workstation.

Operations layer - network operations and protocols

The operations layer is where most people find themselves today. This layer is concerned with defining an enterprise-wide network operating system, and the protocols that will be used in the network. It may sound pessimistic, but there is a lot of ignorance among the sales forces of networking products, and this does not make the task of deciding on what is best for you any easier.

It may be helpful for you to put together a set of questions that you can ask a salesman, to assess the technical competence that they bring to the sales process. Not that non-technical sales people have no value, they have great value, but one of the primary resources that you can draw on, is the technical staff of network vendors. It is worth while getting to know as many techies as possible, so that you can draw on their combined wisdom and knowledge.

However, beware the networking salesman who tells you that his company "..doesn't use TCP/IP, because we use Ethernet instead"! Such a person may not be the best hope for help when you have a problem.

The goal of the operations layer is to allow any individual computer (work-station or mainframe) to exchange data in a comprehensible manner with any other. The base layer was concerned with connectivity, and this layer is concerned with comprehension. These two ideas are important precursors of communication, the task of the Enterprise layer.

Enterprise layer - business needs and strategic concerns

I said earlier that enterprise networking is largely about integrating people into business activities. The Enterprise layer of the three-layer model is concerned with business related issues. Here is where we will find our standards for hardware and software. We will allow our sense of the company's business goals to drive the network at this level.

As client/server computing becomes more prevalent, and the source of a service becomes less relevant, mission critical applications can become truly distributed, and as a result, less vulnerable to any single point of failure.. Other business functions, such as mail (in the form of electronic mail, EDI, or other exotic technical functions) find their place at this level.

Standards & specifications

Standards and specifications are important to an enterprise network. They drive the acceptance of new equipment, and facilitate the repair of broken equipment. As soon as possible you want to begin putting together a list of acceptable work-stations, or other hardware, as well as acceptable software configurations.

In many companies, especially cost conscious companies, acquisition of PCs (which is where most of the variances show up), is done on a current-best-cost basis. However, this can lead to differences in such important components as BIOS chips, hard drives, and network interface cards. These are unglamorous items, but they can have a devastating effect on network performance and reliability. Imagine a situation where 20 different PCs from 20 different vendors all had different software configurations. How can we even begin to track down an elusive intermittent network problem?

In much the same, way, defining a standard for network wiring equipment will allow you to better manage expansion. For example, a standard might say that up to 12 network nodes might be wired back to a wiring closet using Brand A type equipment, but that any increase over that level would require Brand B equipment. By standardizing in this way, you can proactively plan when changes will need to occur.

MANAGING AN ENTERPRISE NETWORK

Mention of such habits as proactive planning leads naturally into a discussion of managing the enterprise network. This is an important component of your original plan, but one that can all to easily be overlooked. After the heady success of designing and

implementing your network, someone is going to have to manage it - and that someone may be you!

This leads to a conceptual problem, because even though you have a mission critical resource, which is vital to the success of your company, you don't have very much to show for it. The personal workstations are the province of their owners/users, and the servers are generally considered as being *separate* from the actual network. Small wonder, then that network management perhaps more so than any other area of MIS management is a source of dyspepsia, depression and divorce!

Management models

We can identify three common styles of managing an enterprise network:

- Central management – the "strong man" model
- Distributed management - the "laissez-faire' model
- Crisis management – the "fire-fighting" approach.

Central management model - "Strong man"

With this style of management, control of the network resides fair and square with the network manager. This person retains total control over the operation of servers, configuration of work-stations, and all other aspects of the network.

Such a scheme can work very well in a technologically naive company, and indeed it is probably the best approach for such an environment. Despite its negative sounding name, it does provide a strong source of support, and allows the importance of the network to the success of the company to be acknowledged.

On the other hand, the potential for problems is enormous. As the level of expertise rises, users are going to want to take charge of their own destiny more and more, and to participate in the management of what they see as *their* resources.

Distributed management model - "Laissez faire"

If the end-users of the network have a good sense of what they wish to accomplish, and have also the necessary sensitivity to the needs of the network, a distributed approach to network management can be a real time saver for the network manager. By placing responsibility for routine network operations like backup, and intra-department moves on the shoulders of end-user management, time is freed up for more demanding tasks.

Beware, however, of adopting this style of management as a political expediency, because the political pressure to comply with this approach can very easily become political pressure to take responsibility for shortcomings on the part of those who were given the control. Distributed network management is best suited to those environments where there is already a strong distributed MIS presence in the form of advocates or in-house

consultants. These people can probably assume some degree of responsibility without too much burden, and the additional duties will represent an interesting challenge.

Crisis management model - "Fire-fighting"

The idea of a crisis management approach to network management may sound like heresy, but there is a case to be made for it. Most of the time things will go along pretty smoothly, and there will be no problems. Networking does not necessarily change everyone's life or business habits. If this is the case for you, then it may make sense to do nothing, and react only when there is a crisis.

A certain amount of semantic *legerdemain* is necessary for this to succeed. Operational activities must proceed backups must be taken, software updates must be installed. However the argument can be made that none of these activities are particularly network oriented, and as such do not constitute network management.

For this approach to succeed, one thing above all is important. There must be a very strong technical resource available on short notice. When a problem arises, fast diagnosis is important to get the network back up and running. If there is no designated network manager, or no technical expertise available, then the idea of 'doing nothing until necessary' will fast become an idea whose time has gone!

WHAT'S IN IT FOR YOU?

Enterprise network implementation and management is still a sufficiently new business activity that it is still largely undefined. It is possible, of course, to regard the introduction of such a network as a simple one-time project, and to continue afterwards with no appreciable change in the business environment. However, this avoids consideration of one of the underlying phenomena of the MIS environment, which is that finally the MIS process is becoming automated.

Take for example, the routine task of delivering a report to the person who requested it. With networks in place to deliver the report electronically to the closest available printer, there will be a declining need for operators to manage large central printers. As much of the way in which computing happens, begins to change, there is an opportunity to help define the way of the future. This opportunity is not for everyone, but it can offer rich rewards both financial and personal for those who take it on, and succeed.

Changing nature of the job

The most significant change is in the very nature of the job. As a typical MIS manager, or technician, your focus is largely on the day-to-day needs of the department; helping to keep the data flowing smoothly, writing programs to improve the situation, or meet a strategic need for one of the end-user departments.

As an enterprise network manager you will become much more involved in the strategic aspects of your company's operations. Rather than dealing primarily with relatively precise needs, which are articulated to meet a specific business goal, you will be far more involved with helping to determine the nature of the goal itself.

People skills

This will require a change in emphasis away from technical skills, towards people skills. Unfortunately, it will also require a large investment in a brand new set of technical skills. The vagaries of people, and the new set of technical skills pose a large demand on your time and inner resiliency! However, if this is what makes you tick, then getting involved with this sort of activity can be very stimulating.

Business awareness

Business activities in this last decade of the 20th century are becoming even more dependent on information than in the past. The days are over, if they ever existed, when technicians could remain isolated from the business needs of their company. In order to help move most businesses into the next century, it will be necessary for those individuals with the capability of abstract and strategic thinking to lend these talents to the enterprise at hand. Fortunately these skills are common among information processing professionals, and if you are willing to make the investment in time, training and turning around their perspective, the challenges of managing an enterprise network may prove to be just what you are looking for.

Growing into NewWave Computing

Jeff Eastman
Thong Pham

Abstract

HP's NewWave Computing strategy offers the prospect of increased productivity for knowledge workers both individually and in teams by providing superior ease of use, automation of complex tasks, and the coordinated sharing of data between different software applications which reside on physically distributed computers. These benefits are realized through the use of such technologies as object-oriented distributed computing, object-oriented software, multi-media user interfaces, 'hot' links, and agents. NewWave Computing applications will initially be installed in environments inhabited by conventional software applications that were not developed using NewWave Computing technologies. Thus there is a clear need for NewWave applications to integrate and interoperate with these applications. This paper will discuss how NewWave Computing software applications can be integrated with traditional, non-NewWave applications, and will illustrate a promising way to ease the transition from current application environments to the NewWave Computing Environment.

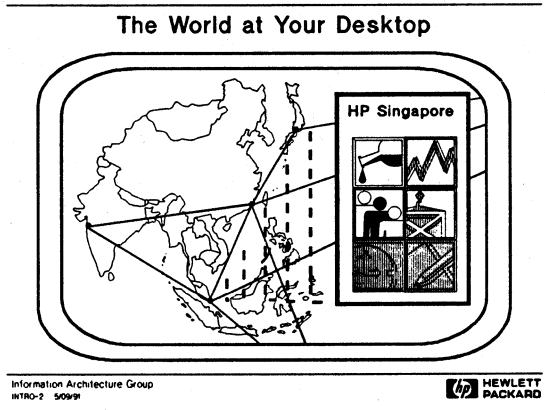
1. Introduction

Based upon the award-winning HP NewWave Environment on the PC platform, HP's NewWave Computing strategy offers the prospect of increased productivity for knowledge workers using a number of computing platforms. Employing such technologies as object-oriented distributed computing, object-oriented software, multi-media user interfaces, dynamic data-passing 'hot' links between applications, and agents, users will be able to achieve superior ease of use, automation of complex tasks, and the coordinated sharing of information between physically distributed sites.

As with any new computing technology, NewWave Computing systems will be phased into user's environments over a period of time. Thus there is a clear need for NewWave applications to integrate and interoperate with the systems and applications which already exist in these environments. This paper begins with a futuristic tour of a NewWave Computing environment which might exist in the timeframe of HP's *1995* vision video, and develops the major architectural elements which will enable such an environment. It then looks at the realities of todays manufacturing systems and discusses the application of the HP Sockets integration product within this legacy environment. Finally, we discuss a promising way to grow these systems into the world of NewWave Computing.

2. What is HP's Vision of NewWave Computing?

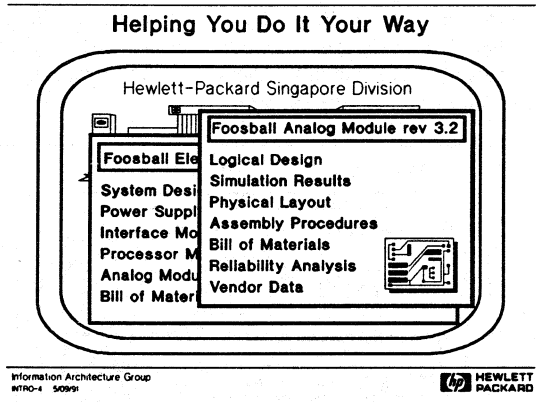
Let's take a tour through the eyes of a typical user of a future NewWave Computing environment: World wide, on-line information services give you 'instant access' to the latest versions of the materials that you need to work accurately and efficiently at your job. By clicking your mouse a few times, you can browse around the system by organization or by geographical area, for example, and 'drill down' to the facility of your choice. Once your access rights have been verified, icons depicting your authorized information domains will be displayed and you may proceed.



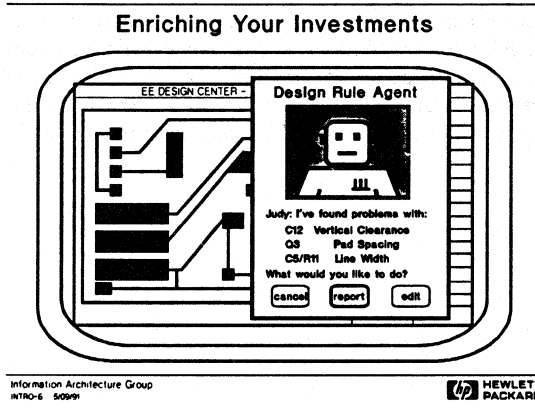
As a user, you will find the on-line browsing facilities to be a natural and intuitive way to access most of the information you need. Related information will be linked together, and you can browse easily along these pathways at the click of a button. There's a natural place for everything in this world, and everything will be found in its place. With a little bit of exploring, you will quickly be able to locate the objects you need and link them into your own work areas. Should browsing fail to locate the information you need, however, just give the system's "finder" a search command and qualifying items will be located for your inspection. You'll need to give the finder some good hints to narrow the search space, though, because a global search will take a very long time to complete.

As you browse through the on-line information, your access rights will be continually checked to verify the portions to which you have authorization, and to advise you of unusual service charges where this is appropriate. Some of your company's most sensitive information will be hidden completely, and you will not even know of its existence. Other information sources will be identified, but you will need to get additional authorization in order to proceed. Still other sources will be made partially

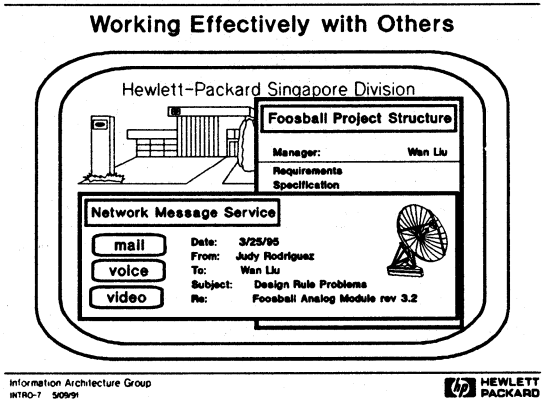
available by limiting the sorts of requests you can make. In your own areas, of course, you will have complete authority to query, view, and customize your parts of the NewWave Computing information world.



Once you locate the information objects which need your attention, the system will automatically select the appropriate tools based upon the nature of your requests. This is done using information which the system maintains on its user's behalf. Many of the tools will be the familiar applications which you've been using for years. These will be integrated with new applications which draw upon the enhanced capabilities of the information base itself. Their combination will be very powerful, allowing you to capitalize upon your current computing investments while being less involved with the low level details of these underlying systems.



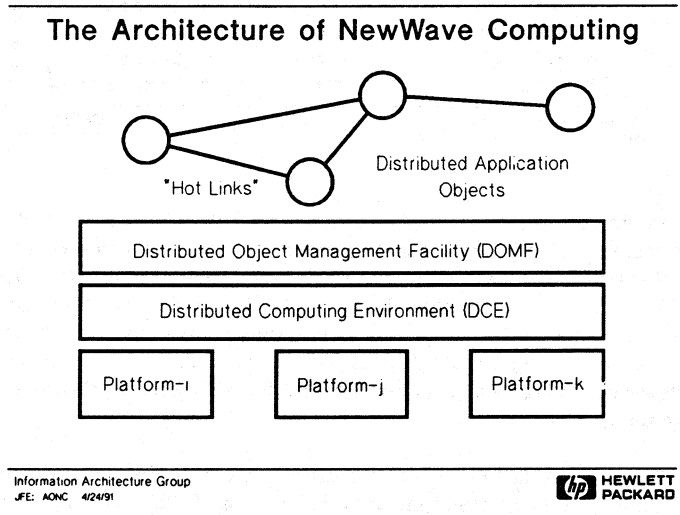
Do you have a repetitive task which you perform often? How about an administrative procedure which you want others to follow? Save time and your valuable creative energy by training one of your agents how to command the system through its paces instead of doing it manually each time. Agents can record your actions as you go through the process once and build a script which they can follow themselves later. Agents can be initiated manually by your requests or automatically when certain events occur. They can communicate with other agents too. Since your agents operate behind the scenes on a 24 hour basis, maybe you won't have to.



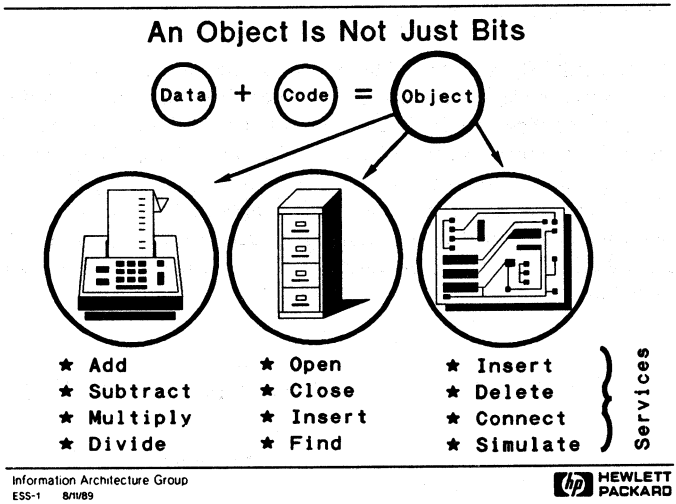
Even the most industrious collection of agents cannot replace a single good employee. With today's decentralized organizations, teams of people spread across many physical sites need instant communications to stay coordinated for maximum productivity. Fortunately, advanced communication facilities are embedded in the NewWave Computing network and are available for your instant use. Simply click a button to open up a text, voice or video link with your counterparts anywhere in the world. If your time zones don't overlap or they are not near their computer nodes, then leave a multi-media message with their mail services. With the latest in portable cellular computers, though, don't be too surprised if you connect right up with your colleague on a fishing trip!

3. The Architecture of NewWave Computing

Sound far-fetched? By now, the reader may be wondering how a system like this could be made to work, so some explanation is in order. We see it constructed from several major elements as shown in the figure below: A multitude of platforms encompassing hardware and operating systems, a unifying distributed computing environment layer which includes all networking services, a distributed object management facility layer, and a growing set of object-oriented distributed applications interconnected with active, 'hot' links. We believe that object-oriented applications will enable the kind of user scenarios presented above. To understand how this is possible, we need to look into object technology more deeply.



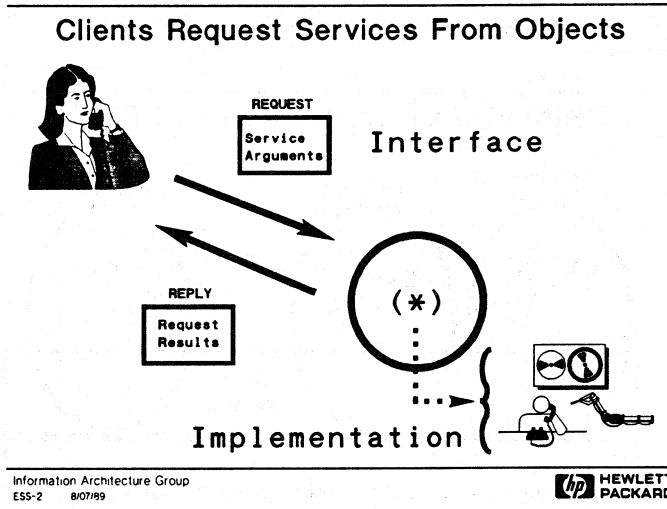
Fortunately, object technology has just a few major concepts which combine to enable a surprising diversity of solutions. The most important concept is the idea that an object is a fusion of data *and* code into an active information entity which embodies the aspects of both. In practice, each object usually represents an abstraction of a useful, real-world, "thing". It may be thought of as a computer simulation of that thing, and this real-world similarity allows the user's intuitions about its uses and behavior to be readily applied to its operation. Various kinds of useful tools, containers, devices and media can be simulated using objects. Objects each make themselves useful by providing a set of services to their user clients.



To get an object to perform one of its services, a client makes a request of that object. Service requests identify one or more objects and/or data elements as arguments and specify the desired service. Once the object has completed this activity, a reply will be returned which may contain other objects and/or data as results. These computations may be very complex, involving interactions with peripherals, non-object computer subsystems, and/or requests to other related objects. In NewWave Computing, object requests are transmitted via the underlying Distributed Computing Environment (DCE) message services.

In the DCE, sets of related services are organized into collections called *interfaces*. An object's interface fully describes the external behavior of that object with respect to all the other objects in the environment. It makes absolutely no claims about the object's implementation, however, which is free to be implemented independently. In this manner, interfaces are said to *encapsulate* their object's implementations. By choosing from an existing set of interfaces, the designer of a new object can select which common services to provide and thereby determine the roles in which the new object can participate with other objects. Once the common services are determined in this manner, the value-added services which make the object unique are added to form the new object's interface. In NewWave Computing, object interfaces are defined in the Class Definition Language (CDL) which formalizes the object's external behavior and hides its internal implementation.

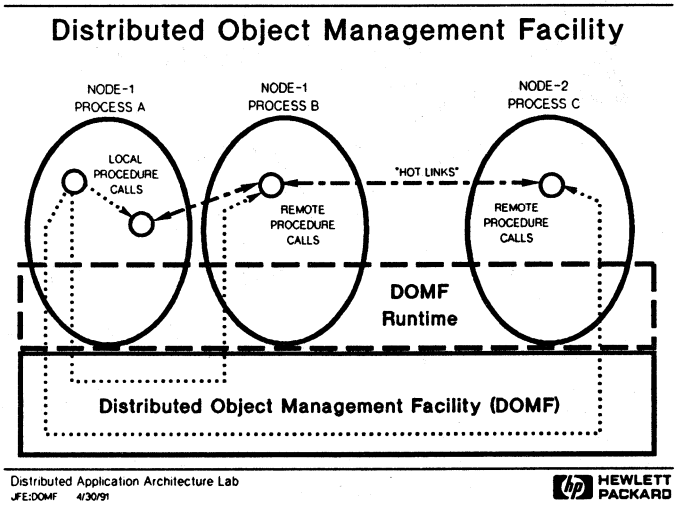
A significant benefit of formalized interfaces is that object service requests can be generic. This means that different objects which support a given interface are indistinguishable in the roles which require that interface. Thus, a client may request the same service of a number of otherwise unrelated objects and the system will be able to select the correct implementation in each case, without further specification. This makes systems of objects extensible after the fact, for an object's implementor need not anticipate all of the kinds of objects which it may encounter in the future. By defining the interfaces which such clients must support, the implementor can be more confident that they will function correctly.



Formalized interfaces also facilitate transparent distributed computing. Within a given implementation of NewWave Computing, sets of objects conforming to various object interfaces may interwork via high-speed procedure calls as in traditional applications. Through their support of standardized interfaces, however, these objects can also interwork with other similar objects on remote systems - without knowing that these objects are remotely located. The following figure illustrates how objects which are physically distributed and which are implemented using different technologies can function as a tightly integrated system.

In this figure, the DOMF provides the basic remote procedure call mechanisms which allow objects to send and receive requests without worrying about the locations of their clients. As it is a distributed facility, the DOMF is actually composed of cooperating object managers which reside at each computing node in the DCE. Each DOMF node provides a set of platform facilities which are tuned to its particular hardware, operating

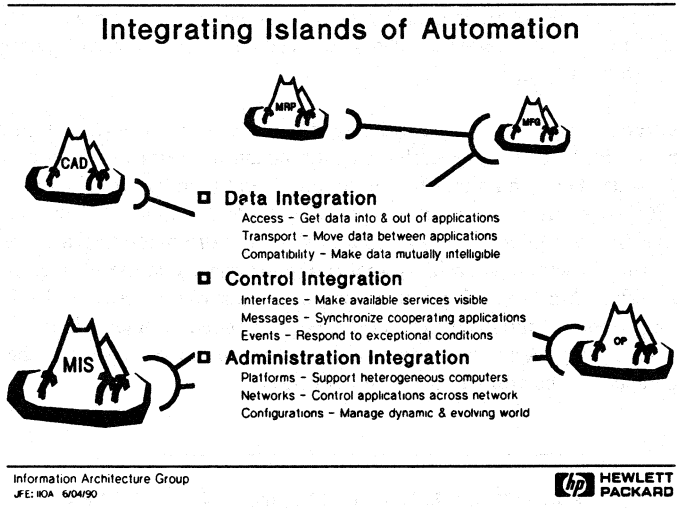
system, and networking environment, plus a set of per-process runtime facilities (DOMF Runtime) which are implemented to match each programming language. The DOMF Runtime is responsible for providing an interface to the platform's core communication services, and for providing implementations of standard policies for common object services and user facilities. By implementing these policies to support a standard set of interfaces on each node, implementors of objects can take advantage of these agreed upon external definitions while also getting maximum advantage of their implementation language's features and capabilities. Of all these policies, the one which supports the management of active "hot" links between objects is probably the most important to achieving our vision of NewWave Computing. These links allow objects to pass information between themselves in a distributed system in a reliable and extensible manner.



NewWave Computing promotes distributed processing, resource and data sharing through the use of an object-oriented interface methodology. This methodology facilitates the building of cooperative environments where applications view each other as objects or as sets of objects. All interactions between objects are channeled through service request messages delivered locally or remotely by the DOMF. The data structures of each application is completely encapsulated and changes made to one object will have minimum or no effect on other objects in the community. Because of this encapsulation, implementors are free to choose the most appropriate programming language for their application; either object-oriented languages such as C++ or Smalltalk, or conventional, non-object-oriented languages such as C, Pascal, Fortran, and COBOL.

4. NewWave Computing and Conventional Software Applications

Todays computing environments are dominated by a legacy of heterogeneous platforms supporting an archipelago of file and data based "islands of computing" which have evolved over the years. The recent introduction of computer networks into this environment has created the potential for integrating these application centers, but significant problems remain.



Conventional software applications were often developed to solve specific problems without concern for global data or resource sharing. Most of today's conventional applications are not distributed. They were developed with a traditional programming methodology which resulted in a hard-coded and highly optimized solution of the problem being solved. Because these applications address the specific business needs which were apparent at the time of their genesis, they often ignore the needs of other business functions which have become more important during their lifetimes. Thus, as the problem changes, or as its scope of usage is broadened, the application's construction often collapses. Finally, since conventional applications were not developed with the objective of interacting with each other, integrating them is usually expensive and time consuming.

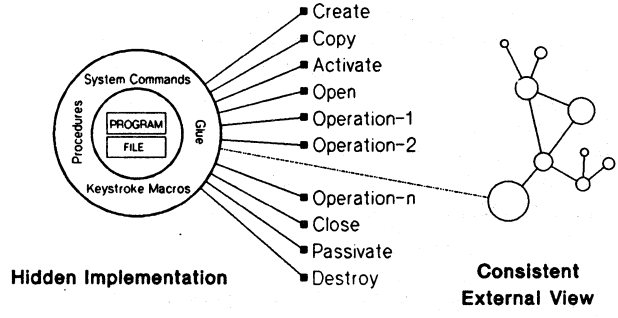
The non-cooperative aspect of conventional applications has become a dilemma for systems administrators as systems integration has become necessary in order for corporations to survive in the increasingly competitive world market. To cope with this problem, systems integrators have looked for software tools to do their tasks faster and

more cost effectively. New software development environments have been invented, software tools and frameworks have been created, and new computing paradigms have been conceived and promoted. NewWave Computing is one example of this trend. Although the benefits of NewWave Computing are well recognized, customers cannot in an instant forgo all of their existing applications. A working program which solves a current business problem is too precious to discard and is often impossible to reproduce or extend. The challenge for customers is how to make a smooth transition from their current legacy of applications and systems to the NewWave Computing environment without interruption of their business operations and with minimum cost. Hewlett-Packard is aware of this important customer need and has been working hard to find an answer for this challenge.

One powerful way to integrate existing applications is through the encapsulation inherent in object technology itself. By formalizing the services which an application provides to its clients in an interface description, and then by providing a procedure to access each service, any application can be made to 'look like' an object. HP's NewWave Bridge Builder is an example of this kind of embedding technology on the PC platform. By wrapping ordinary application programs and the files upon which they operate with a thin layer of system commands, keystroke macros, and other 'glue', it creates a procedural interface for a new class of embedded objects in its environment. These objects may then participate as citizens with the other objects in the NewWave desktop environment without further modification.

EMBEDDING EXISTING PROGRAMS

Another NewWave Contribution



Information Architecture Group
JFE: EEP 4/30/91

HEWLETT PACKARD

By extending this approach, the level of an application's integration with the rest of the

object world can be significantly improved. This requires reworking the application to support some of the most important interfaces directly, and may involve considerably more effort. While achieving the full power of HP NewWave requires fully integrated object implementations, a small number of true HP NewWave objects, such as the Folder and Write objects, augmented by a variety of embedded applications, have created an environment which is truly revolutionary in its ability to organize, manipulate, and present information.

5. A Growth Path through HPSockets

Another example of object-oriented embedding technology is provided by the HP Sockets product. Unrelated to the Unix Sockets mechanism, HP Sockets is a software tool designed specifically for the integration of existing applications in a network of heterogeneous computers. HP Sockets facilitates the application integration task by reconciling differences between various computer platforms and by making data accessible to the applications without forcing changes to either the producer or the consumer of that data. HP Sockets provides functionality which encompasses that of:

1. message and file transfer systems
2. data manipulation and translation systems
3. local and remote program control systems
4. remote procedure call systems

HP Sockets also provides network-wide system management and logging of data and errors to ensure the reliability, auditability and recoverability of an integrated software system. Its program control functionality allows automatic start up or shut down of participating applications so that a system can function harmoniously as an integrated whole with minimal human intervention.

The data manipulation and translation capabilities of HP Sockets preserves the data models of existing applications. Data types and structures can be different at each end of a message path, thus a participating application in an integrated system can be replaced without affecting the rest of the applications and the replacement can be done without modification of the application code. Any changes in data elements, data types or message formats only requires a reconfiguration of the system, not recoding of an application program.

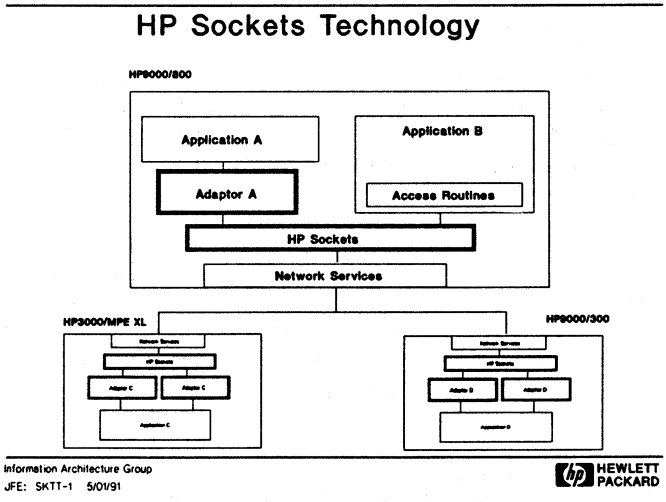
HP Sockets currently runs on HP9000 computers, series 800, 700, 600, 400 and 300 and on HP3000 computers series 900. A server is also available to support connections to IBM/MVS. Investigations are being made to port HP Sockets to additional, non-HP, platforms.

6. How An Integrated System Functions In An HP Sockets Environment

In an HP Sockets environment, applications are shielded from changes because they are allowed to keep their own data models while operating as part of an integrated system. Each application receives data in formats required for its processing and produces data as if it were operating in its stand-alone mode. This preservation of data models is achieved by using the data manipulation capability of HP Sockets and by encapsulating the application with a software layer called the *application adaptor.* An application adaptor defines an interface between the application and HP Sockets. It:

1- Gets data into or out of the application data storage (memory, data base, or files).
2.- Sends/receives data to/from other applications via the network.
3.- Starts, stops and controls other processes running within the environment.

In a HP Sockets environment each component application can operate synchronously or asynchronously. This flexibility is provided through the sophisticated messaging mechanism of HP Sockets. Applications can focus on their own processing until an external request is received or until there is a need to communicate with the external world. A component of an integrated system can be off-line until it is invoked by another component when its processing is needed.



HP Sockets is designed to integrate existing applications running in current manufacturing environments. In the next section, we will show how it can also be used as a gateway to the new world of NewWave Computing.

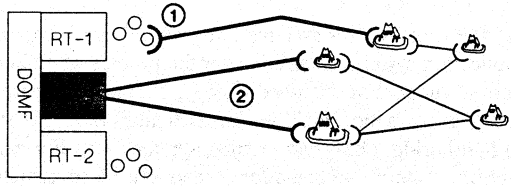
7. Integrating Conventional Applications Into the NewWave Environment

To operate in the NewWave Computing environment, a conventional application must export an interface which defines the services it can provide, the requests it can accept, and the interfaces it expects for the arguments of its requests. Since HP Sockets application adaptors facilitate the encapsulation of applications by defining message and data passing interfaces between them, it is also a natural way to begin to embed an application as an object within the NewWave Computing community. Even though the adaptors are written in conventional programming languages and for conventional application programs, their function is very similar to that of an object's interface. Each adaptor receives a service request from an outside client, processes the request and returns results to the requester. Adaptors can also initiate service requests and handle returned results. Thus, it is natural to think about an adaptor design in the object-oriented paradigm.

HP Sockets Can Talk to NewWave Objects

① Individual "Ambassador Objects" constructed in an existing DOMF Run-time environment can act as an adaptor as well to access a single legacy application

② An HP Sockets Run-time interface to the DOMF could provide a general purpose NewWave computing adaptor, allowing for direct communication between application adaptors and the DOMF communication core



Information Architecture Group
JFE: SCA-1 4/29/91

HEWLETT PACKARD

The major problem in embedding a conventional application comes from the need to define a formal interface between the encapsulated application object and the other objects in the NewWave environment. With the application adaptor, this interface can be built at either end of an HP Sockets communication path. The preceeding figure illustrates both alternatives: 1) The use of existing application adaptors with newly-created "ambassador objects" and their adaptors; 2) The use of a general purpose HP Sockets Run-time interface to the DOMF which could provide a generic ambassador service for all application adaptors.

The first example is the most straightforward. Here, an actual NewWave object must be built to represent the existing adaptor to the NewWave environment. This object acts as an ambassador to access the application's adaptor from the NewWave environment. It accepts request from other objects in the environment, sends these requests to the adaptor via its own HP Sockets adaptor, receives returned results, and gives the results to its requesters as if it has produced the results by itself. The ambassador object can also take a request from the adaptor, send it out to another object in the NewWave environment, receive the desired result and send the it back to the adaptor via HP Sockets. To other objects in the NewWave environment, the ambassador object is just a regular object. Since the other objects do not need to know how their requests will be processed by the ambassador object, they don't know about the conventional application behind the scenes. They are not aware of the adaptor interface to the legacy application and they don't need to be. In brief, an ambassador object provides services of a conventional application by making itself a virtual adaptor for that application in the NewWave environment.

While the use of ambassador objects with existing application adaptors is simple to describe, it places all of the implementation burden on the systems integrator. A more powerful solution would be to develop a tightly coupled run-time interface between HP Sockets and the NewWave DOMF. This interface would provide the equivalent of a generic ambassador object, but directly connected to the DOMF. This would allow systems integrators to concentrate on defining their end of the communication path, and would tie the two worlds together at the lowest possible level of the architecture. We think this approach is very promising.

In any approach, the first step for systems integrators is to formalize the interfaces of their existing application systems so as to begin the process of embedding. HP Sockets is an effective tool in support of this need which is available immediately. Its rich data manipulation and transformation facilities, in particular, offer the systems integrator a powerful tool to build bridges between existing applications. For early integration with NewWave Computing systems, ambassador objects can be constructed which use these adaptors to move information into and out of the NewWave environment.

8. Conclusion

This paper has discussed a evolutionary path which we believe will enable customers to smoothly integrate their existing applications into the evolving new world of NewWave Computing. It stresses the use of HP Sockets as an integration tool between existing applications today, as an embedding mechanism to formalize the interface definitions for these systems, and as a data manipulation mechanism for integrating with future NewWave Computing object-oriented applications as they arrive.

4113
Developing Applications with Client/Server ALLBASE

Anjali P. Mulgaonkar

Commercial Systems Division
Hewlett-Packard Company
19447 Pruneridge Ave.
Cupertino, CA 95014

Abstract

Client/Server applications are being rapidly adopted in the industry due to the productivity that they bring end users through increased ease of use and flexibility. This is because Client/Server applications make full use of graphical user interfaces like MS Windows 3.0 for end user presentation services, while relying on the OLTP capabilities of the server to manage large amounts of data. Client/Server ALLBASE is a software product that provides a PC-based interface to the ALLBASE/SQL database on an HP 3000 or HP 9000 server. This paper presents an overview of the internals of this product, the new application programming interfaces and the tools that are available to develop Client/Server applications using HP's ALLBASE/SQL.

What Is Client/Server?

A client/server architecture is one in which client applications executing on desktop computers cooperate with servers executing on back-end computer systems to perform a specific function. The concept is based on splitting the application into two pieces—a front-end portion on a desktop computer, which handles local data manipulation and user interface, and a back-end portion on a server, which handles database and other transaction intensive processing. The server can be a PC, a minicomputer, or a mainframe depending on the demands of the application. Likewise, the client platform can be scaled and positioned according to the individual user's desktop and application needs.

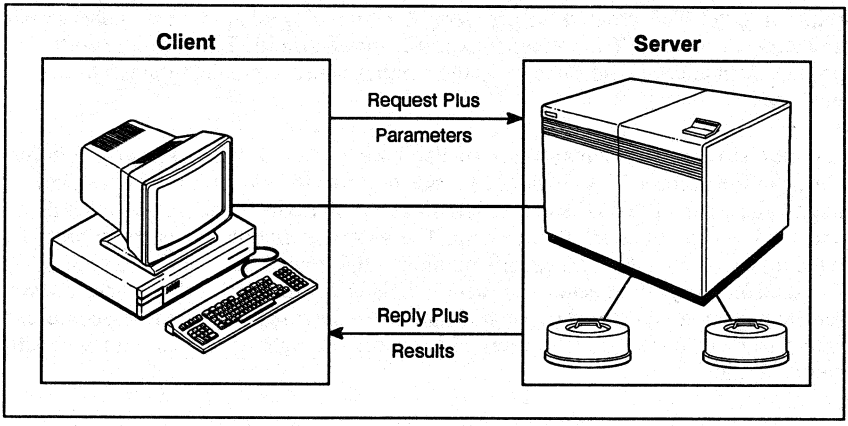


Figure 1. Client/Server Defined

A study of Fortune 1000 MIS departments by Boston-based Yankee Group Inc. shows that 69 percent of the respondents have client/server systems or plan to implement them soon—85 percent of these use client/server for mission-critical applications. These numbers are bound to increase due to the flexibility and increased productivity of the underlying technologies.

Why Client/Server?

The client/server architecture evolved out of the changes in the needs of the organizations that use computers and the rapid growth in the technology available for computing. A large mainframe, responsible for satisfying all the computing needs of the organization had its limitations, which were gradually becoming unacceptable. As the organizations moved towards decentralizing the decision making process, an acute need was felt to have the data available at dispersed locations. Also, with the data and computing centered at the mainframe, the poor response time was becoming a major issue. The mainframe applications had their user interface supplied by a dumb terminal, which was quite restricted, being only character based. As the need for diverse range of information services increased, the user needed to surround himself with a number of terminals to access these various systems.

At about the same time, the technology advances were bringing down the cost of processors and memory. Personal computers and workstations started appearing on people's desks. The PCs, inherently, are capable of displaying bitmap graphics. This facilitates the use of color, scroll bars, pull down menus, popup windows, and a variety of other graphical objects for creating an easy-to-use and easy-to-learn user interface. Programmers now had access to tools like MS Windows and OS/2 Presentation Manager to allow them to fashion their application's user interface to their specific needs. The advances in network technology made it cost-effective to connect the geographically dispersed computers together—and to do so reliably.

The central computer, could now be used as a "file server". When a file server runs on a LAN, all application processing happens on the user's PC, while the server maintains the data files. A file server sends the entire file across the network to the requesting PC. This created the problem of maintaining adequate user concurrency and data consistency. The tremendous traffic burden on the LAN, often resulted in poor performance and bottlenecks as the number of users making requests to the file increased.

A solution to these problems was seen in database servers. Database servers are different from file servers. The client (PC) sends requests for data across the network to the database server that is listening. The server receives the requests, processes them, and sends the results back to the client. The server sends only the parts (subsets) of the database requested. This greatly minimizes the flow of information on the LAN. The data integrity features like transaction logging, rollforward and rollback, and system management built into large mainframe and mini-computer databases; combined with high input/output transfer rates and network throughput, makes this a very efficient solution.

Client/Server and Relational Databases

A relational database system has the flexibility and functionality needed to work across networked computer systems. The front-end application on the client uses SQL to talk to the back-end database engine on the server. The client is responsible for executing the application. The client formats the SQL command and sends it across the network to the database server, which then parses and executes the SQL command and sends the result back to the client. Most database servers use SQL because it is a convenient language for specifying logical subsets of data. SQL standards are also specified by both the American National Standards Institute (ANSI) and the International Standards Organization (ISO). These bodies specify the standard for embedded SQL. A standard SQL call-level interface is currently being developed by the SQL Access Group.

Client/Server ALLBASE

Client/Server ALLBASE is a software product that provides a PC-based interface to the ALLBASE/SQL database on an HP 3000 or HP 9000 server. It provides a client/server programmatic interface to ALLBASE/SQL to enable the most rapid porting and development of leading client/server applications and tools. This will enable Windows 3.0 applications developed with Gupta technologies SQLBase/C Application Programming Interface (API) version 4.1, to access ALLBASE/SQL database with little or no change. The architecture of Client/Server ALLBASE allows these applications to simultaneously access other databases on their servers. In the future, we will also support the call-level SQL API from the SQL Access Group, once it has been finalized.

HP's Client/Server ALLBASE Architecture

The client portion of the product consists of Gupta SQLBase/C API Dynamic Link Library (DLL), plus a router for ALLBASE from Hewlett-Packard. The router communicates with the API by windows messages using the message-level interface (MLI) format. The router translates the MLI messages to SQLX packets and transmits them over the LAN to the ALLBASE distributed execution agent (DEA). Gupta's DLL supports an MS-DOS client environment running Microsoft Windows 3.0. Hewlett-Packard will support MS Windows 3.0 clients at the first release.

The server portion of the product consists of the ALLBASE distributed execution agent and listener processes for Netware and NetIPC.

Communication between the client and the server is accomplished with messages. A requester of services sends a request message to an appropriate server, and the server sends back a response message. Each request demands a single response. All response messages are synchronous with their corresponding requests. A requester cannot send subsequent request messages without first receiving a response from the server for the last message issued, and a server cannot send a response message without first having received a request. The request messages are generated as a consequence of an application program issuing one or more Gupta SQLBase C/API function calls.

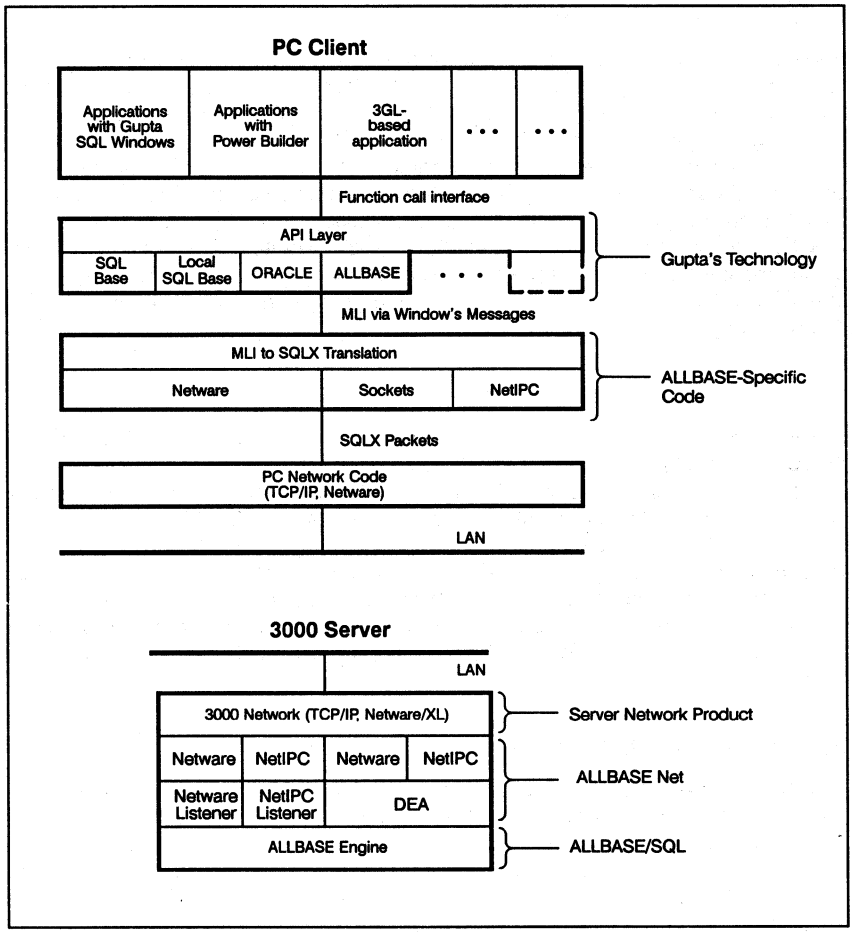


Figure 2. Client/Server ALLBASE Architecture

What Is ALLBASE/NET?

ALLBASE/NET allows an application on a local machine (the client node) to access an ALLBASE/SQL DBEnvironment on another HP machine (the server node), where the the client and server are connected by a LAN or a WAN. The ALLBASE/NET product is composed of several parts. On the client node is the AliasDB file, and on the server node is the ALLBASE/SQL back-end process, the NETUsers file, and the listener daemon.

The listener daemon serves two important functions. It listens for connection requests, and it sets up a direct communication line between the user and the database process. The listener monitors a specified datacommunications port for any incoming connec-

tion requests. When it receives one, it determines the identity of the client user and spawns a database process for that user. After spawning the database process, the listener redirects communications so that the user and the database process communicate directly with one another over the network, thus leaving the listener daemon free to resume listening for more connection requests. There is a listener daemon for each different kind of network service.

The NETUsers file, whose main purpose is security, resides on the server. The file contains an entry for each user that is permitted access to a DBEnvironment on the server node. An entry in the NETUsers file consists of the client node name, the login name of the user trying to gain access, and the login name to be used on the server node when accessing the DBEnvironment.

The AliasDB file resides on the client node. It contains an entry for each aliased DBEnvironment that can be accessed remotely from that node. An entry consists of the alias name, the node name of the server, the real name and location of the DBEnvironment on the server node, the type of machine that the server is running on, and the type of data communication protocol being used. This information is needed to set up the communication link between the client and the server machines and to make the connection to the remote database.

What Is the Gupta MLI?

The message layer interface (MLI), is Gupta's message protocol used between Gupta's servers and clients, as well as internally between Gupta's API and its Routers. The MLI is a request reply protocol in which each request sent by a requester (for example, the client) requires a reply from the server.

Request packets are byte streams, each composed of a header and an optional body. The header contains a code that specifies the function to be executed and a cursor number, which identifies an instance of a database connect. If the requested function has parameters, these are present in the body of the packet in an agreed upon format.

Reply packets are byte streams, each composed of a header and an optional body. The header contains a return code. If the code code = 0, then the requested function completed successfully; otherwise, the code is the error number associated with the failure. Some functions return data to the requester. In this case, the body contains this data.

What Is SQLX?

SQLX is both a function call interface (ACI or API) to the HP ALLBASE database and a protocol for the communication of ALLBASE distributed execution agents (DEAs) developed by Hewlett-Packard.

The SQLX function call interface is a single entry point interface present on MPE XL and HP-UX systems. The caller specifies the function to be performed using a code parameter, and the data needed by the given function code is passed in formatted control blocks passed as two additional parameters to the SQLX function.

The SQLX protocol enables the DEAs to communicate. The client in Client/Server ALLBASE uses this protocol to talk with the DEA on the server. The SQLX protocol is a request reply protocol, where each request expects a reply. The packets are simply a serialized form of the function code and control blocks that are normally passed to the SQLX function.

What Is the Router?

At its simplest level, a router forwards API function requests in a suitable format to the router's respondent database. The router also captures the replies from the DBMS and returns them in a form that the API can accept.

The router for HP ALLBASE/SQL maps the Gupta function codes to the SQLX calls. This not only involves call mapping, but also involves translating Gupta-specific SQL constructs to the equivalent ALLBASE/SQL construct. For example, numbered bind variables in the SQL statements (:1, :2) are converted to dynamic and static bind variables depending on the statement type. Another example would be addition of semicolons at the end of SQL statements as ALLBASE/SQL requires.

The router communicates with the upper layers of the Gupta's API using Windows messages that are formatted according to the MLI protool.

What Is a DLL?

Microsoft Windows provides special libraries, called "dynamic link libraries" (DLLs) that let applications share code and resources. Windows uses DLLs to provide code and resources that all Windows applications can use. In addition, one can create one's own DLLs to share code and resources among one's applications.

A DLL is an executable module containing functions that Windows applications can call in order to perform useful tasks. DLLs are similar to run-time libraries. The DLLs are linked with the application at run-time (dynamic linking), not when the application files are linked using the linker (static linking). In static linking, the linker copies the routine from the static link library to the application's .EXE file whenever the application uses it. If two applications are running simultaneously and they use the same static-library routine, there will be two copies of that routine present in the system. DLLs, on the other hand, being dyanamic link libraries, allow several applications to share a single copy of a routine. Import libraries are used to resolve references to external routines. When the application uses a routine from a DLL, the linker does not copy any code. It copies information from the import library, which indicates where to find the desired code in the DLL at run time. During application execution, this relocation information creates a dynamic link between the executing application and the DLL.

The Supported Networks

For the network connection between the client PCs and the HP 3000 and HP 9000 servers, both TCP/IP and Netware LAN protocols will be supported.

The Supported Clients

The supported client platform will be Intel-based PCs running under Microsoft Windows 3.0 in the DOS environment.

The Supported Servers

The supported server platform will be the HP 3000 running under MPE XL and HP 9000 under HP-UX.

The Supported Gupta API Function Calls

As stated earlier, the client software consists of Gupta C/API and the HP router for ALLBASE. The API calls supported by Hewlett-Packard are those supported for the Gupta router products. The following subset of Gupta SQLBase C/API function calls will be supported:

Call List

sqlbld – Bind long data by name
sqlbln – Bind long data by number
sqlbnd – Bind data by name
sqlbnn – Bind data by number
sqlcbv – Clear bind variables
sqlcex – Compile and execute an SQL statement
sqlcmt – Commit work
sqlcnc – Connect to database or cursor
sqlcom – Compile an SQL statement
sqpcrs – Close restriction mode and result set mode
sqlcty – Return command type
sqldes – Describe items in an SQL select statement
sqldir – Return a list of databases
sqldis – Disconnect from cursor or database
sqldon – Deallocate Windows resources
sqldsc – Describe items in an SQL select statement
sqlelo – End long operation
sqlepo – Get the error position within the statement
sqlerr – Get error text for given error number
sqlexe – Execute a compiled SQL statement
sqlfer – Get full error text for given error number
sqlfet – Fetch the row at the current cursor position
sqlget – Get database attributes
sqlgfi – Get fetch information
sqlgls – Get long size
sqlims – Set input message size
sqlini – Initialize API dynamic link library under MS Windows
sqllsk – Long seek
sqlnbv – Return the number of bind variables
sqlnrr – Return the number of rows in the result set
sqlnsi – Get number of select items

sqloms – Set output message size
sqlprs – Position in the result set
sqlrbf – Return rollback status
sqlrbk – Rollback work
sqlrcd – Return the condition for the most recent call
sqlrlo – Read long operation
sqlrow – Number of rows affected by the last operation
sqlscn – Set cursor name
sqlset – Set database attributes
sqlsil – Set isolation level
sqlsrs – Start restriction and result set mode
sqlssb – Set select buffers
sqltec – Translate error code
sqlwlo – Write long operation
sqlxad – Add two numbers in Gupta numeric format
sqlxcn – Covert a string to Gupta numeric format
sqlxda – Add two dates
sqlxdp – Covert a date to picture format
sqlxdv – Divide two Gupta numerics
sqlxml – Multiply two Gupta numerics
sqlxnp – Convert Gupta numerics to picture format
sqlxsb – Subtract two Gupta numerics

The client/server applications can be developed by two different methods:

- using the 3GL C
- using the 4GL tools

Application Development Using C Language

You can create applications by writing to the APIs directly, using the Microsoft C language. The programmer also needs to be well versed in the Microsoft Windows application development environment, as well as the Gupta SQLBase C/API. A list of the APIs supported by Hewlett-Packard is listed above.

An Example of Programming

To access the ALLBASE/SQL database on the server, the following three steps have to be taken.

- Connect the client application to the database on the server,
- Perform operations on the database,
- Disconnect from the database.

The pseudo-code sample that follows has been taken from the "SQLBase C Application Programming Interface" reference manual from Gupta Technologies, and the reader is referred to this manual for further detail. There are differences between the ALLBASE/SQL database management system and the SQLBase database management system that cannot be totally hidden from the application programmer who is writing applications in this fashion. These differences are in the following areas:

- data types
- SQL syntax
- naming conventions in system catalogs and views
- error return values
- error messages

The reader is referred to the client/server ALLBASE manuals from Hewlett-Packard for a detailed description of these differences.

Connecting and Disconnecting from the Database

```
#include "sql.h"

int PASCAL WinMain (hInstance,hPrevInstance, lpszCmdLine, cmdShow)

HANDLE        hInstance;          /* current instance */
HANDLE        hPrevInstance;      /* previous instance */
LPSTR         lpszCmdLine;        /* command line */
int           cmdShow;            /* to show window or icon */

{

short   ReturnCode;

/* Before the first database connect, you need to initialize the dynamic
    library used for a Microsoft Windows application using the SQLBase
C/API
   sqlini. MS Windows MakeProcInstance function needs to be used to create
    the callback function pointer. This function is called to yield to
other
   MS Windows applications when the server is processing a request. */

if (ReturnCode = sqlini (MakeProcInstance (YieldProc, hInstance)))
   {
    prints ("Cannot initialize API interface - %u\n",ReturnCode);
    return FALSE;
   }

int FAR PASCAL YieldProc ()
   {

    MSG    msg;

    while (PeekMessage (&msg, NULL, 0, 0, PM_REMOVE))
          {
            TranslateMessage (&msg);
            DispatchMessage  (&msg);
          }
    return TRUE;    /* return successfully */
   }
}
```

```
/* Now connect to the database named "sampledb" for the user "sampleuser"
   with password "samplepwd". */

SQLTCUR     cur;      /* cursor handle */
SQLTRCD     ReturnCode /* return code */

{
if (ReturnCode = sqlcnc (&cur, "Sampledb/sampleuser/samplepwd",0)
   {
     printf ("Failure on connect (ReturnCode = %d) \n",ReturnCode);
     exit (0);
   }
else
     printf ("Connection established\n");
}

******* Perform database operations here **********

/* Disconnect from the database now. */

{
if (ReturnCode = sqldis (cur))
   {
     printf ("Failure on DISCONNECT (ReturnCode = %d) \n",ReturnCode);
     exit (0);
   }
else
     printf ("Disconnected successfully\n");
}
```

An Example of Database operation

INSERT with Bind Variables

This example shows how an INSERT command with Bind Variables is performed with the SQLBASE C/API. This example reads input from the user that consists of a row with four column values on a line. Each column value is separated with a comma.

The INSERT command is

```
INSERT INTO X(A, B, C, D) VALUES (:1, :2, :3, :4)
```

The general flow is
1. compile the SQL command,
2. bind the data,
3. execute the SQL command.

```
#include "sql.h"
#include "stdio.h"

SQLTCUR       cur = 0;     /* cursor number      */
SQLTDAP       cp;          /* character pointer */
SQLTDAL       length;      /* length             */
SQLTBNN       bnn;         /* bind number        */
```

```
char          line[80];   /* input line          */
SQLTRCD       ReturnCode  /* return code         */

static   char inscom[] = "INSERT INTO X(A, B, C, D) VALUES (:1, :2, :3,
:4)";

/* Compile the insert command */

if (ReturnCode = sqlcom (cur, inscom, 0))
   {
    printf (" Failure on COMPILE (ReturnCode = %d)\n",ReturnCode);
    exit(0);
   }

/* Insert the data */

Read into the variable "line", the input string.

for (cp = line, bnn = 1; bnn <= 4; bnn++)          /* scan the line */

    {
     length = strspn (cp, ',');                    /* locate comma  */

     if (ReturnCode = sqlbnn (cur, bnn, cp, length, 0, SQLPBUF));
        {
         printf ("Failure on BIND (bnn = %d) (ReturnCode = %d)\n",
                  bnn, ReturnCode);
         exit(0);
        }

     cp + = length;
     if (*cp == ',')
        cp++;
    }

if (ReturnCode = sqlexe(cur))          /* execute the INSERT command */
   {
    printf ("Failure on EXECUTE (ReturnCode = %d)\n",ReturnCode);
    exit(0);
   }
```

NOTE: You will need to add the Windows specific code to the examples above to bring up the appropriate Windows, display information to the user and gather his input. The error handling will also have to be modified accordingly.

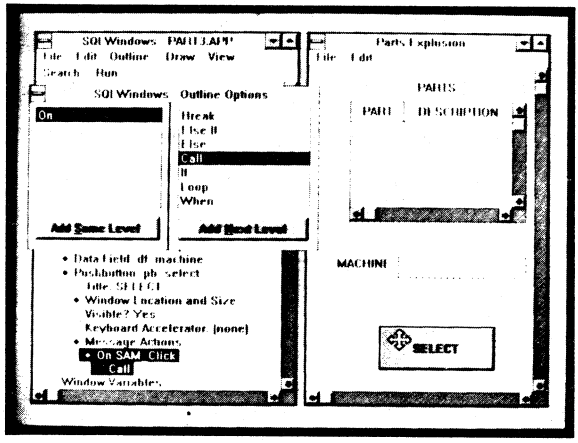
Application Development Using the 4GL Tools

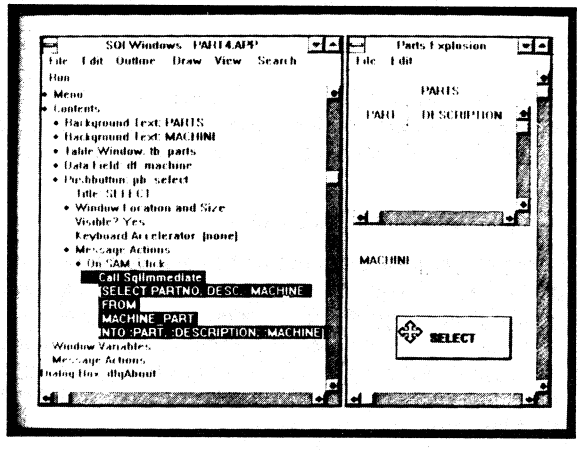
Note that in the example shown above, developing applications by writing to the APIs with the Windows software involves a major investment in time and programmer's skill. To do this job, not only is a proficient C programmer required, but this programmer also must have mastered Windows programming itself and must have a good knowledge of the Gupta's APIs. Currently, there are two application development tools available that let you deliver production applications for ALLBASE in much less time and with a lot less programming expertise:

- SQLWindows from Gupta Technologies
- PowerBuilder from PowerSoft.

SQLWindows

Gupta's SQLWindows was voted the Best SQL Product of 1990 by the readers of Data Based Advisor Magazine and the best "Front-End" and "Windows Database" product for 1990 by the readers of DBMS Magazine. SQLWindows is for professional developers and entry-level programmers who want to build graphical SQL applications that integrate the point_and_click simplicity of Microsoft Windows with the power of SQL. SQL Windows applications can access SQL databases throughout an enterprise, including mini and mainframe DBMSs (like ALLBASE/SQL, Oracle, and DB2) and PC LAN database servers (such as Gupta's SQLBase Server, IBM's Extended Edition Data Manager, and Microsoft SQL Server) all from within the same application.





Full-Featured Toolset

SQLWindows is a full-featured toolset that includes an integrated forms designer, a full-featured 4GL, a powerful application generator, a graphical report writer, and a single-user version of Gupta's SQLBase server. No knowledge of Windows API or C is necessary to create next-generation applications.

The powerful SQLWindows Forms Designer looks like a drawing tool, yet works like an application generator. Code is automatically generated in the Application Outline as you draw visual items in the Design Window. SQLWindows Applications Language (SAL), the 4GL included, is an easy, concise, and powerful set of programming statements. Incorporating SAL function calls in your Application Outline turns SQLWindows Forms into powerful database applications. Express Windows is an application generator that lets you automatically produce several types of data entry and query applications that access your SQL databases—without knowing SQL or SAL. Express Windows works hand-in-hand with Express Edit, an application customizing tool that lets you design the visual interface that you need for your applications. Report Windows is a graphical report writer that works like the latest graphical word processors for designing reports, but is integrated with SQLWindows for access to your data. The open architecture of SQLWindows also allows you to call from your SQLWindows applications the external routines written in C and functions from Windows.

Developers can use the single-user version of SQLBase to develop and test SQLWindows applications without a LAN. For program testing, SQLWindows can be easily switched from design mode into run (user) mode with a single mouse click. Using the animation facility, code is highlighted in the outline as the application runs. Step-by-step and break-point execution modes are provided to further SQLWindows code verification. When an error occurs, the line of code that caused the program to fail is automatically highlighted in the outline. Once development is complete, an application can be saved as text or as a compiled executable version. Compiled run-time applications can then be distributed with SQLWindows run time to end users to access ALLBASE/SQL and other SQL databases throughout the enterprise.

PowerBuilder

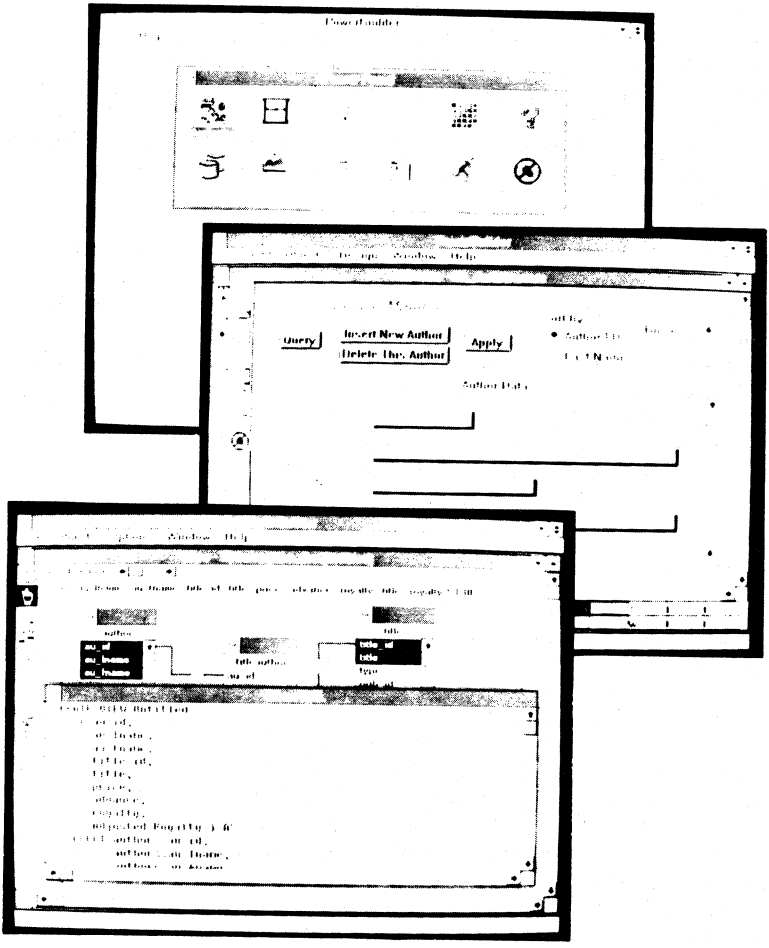
PowerBuilder is Powersoft Corporation's graphical, PC-based client/server applications development environment. Designed for the commercial application software developer, PowerBuilder provides comprehensive and robust development tools in a PC-based, mouse-driven, point-and-click environment for graphically building client/server applications. PowerBuilder has a full suite of graphic painter tools, a high-level 4GL called PowerScript, which is a object-oriented language, and an integrated development and database administration environment. All of these are built on a shared object repository and shared user-defined libraries that store application objects. PowerBuilder also allows you to integrate with the PC environment using dynamic data exchange (DDE), to interact with other Windows applications and file import/export capabilities in a variety of PC file formats. The ability to call C functions in Windows dynamic link libraries is also provided. Hypertext help facilities are available for the developer and the PowerBuilder application user.

Application Objects

PowerBuilder applications consist of two major types of objects: windows and menus.

You create PowerBuilder application windows with the Window Painter. Within a window you can place any of the objects that MS Windows supports and several additional objects that allow you to enhance the window's appearance and provide automatic facilities for database access and update. Most objects can respond to events in the Windows environment by executing PowerScript routines that you write to define how the objects should respond to the events.

A *menu* is an independent PowerBuilder object that you create in the Menu Painter. You can organize menus into any number of levels, including cascading menus. Like



objects within Windows, menu entries also respond to a clicked event by executing a PowerScript routine that you write. You can also associate a menu with a window.

PowerScript contains more than 200 commands or functions for manipulating the graphic objects in your application, for performing calculations, for doing conditional execution, for looping, and so on. It also provides relational database processing through embedded SQL statements for both cursored and non-cursored SQL operations, as well as for stored procedures in the DBMSs that support them. Compared to traditional Windows application development in C, PowerScript provides a very concise language for defining your processing requirements.

Using the PowerBuilder Application Painter, you give each new PowerBuilder application a name, assign application defaults such as type styles and sizes, and specify an icon that will identify the application to its users. When a user starts a PowerBuilder application, the system "opens" the application and executes the PowerScript for its "open event". You write this script in the application painter, establishing values for the application's variables and opening the first window that the application will display. Other events for which you can develop application-level scripts include the idle event and a system error event.

A *data window* is a special type of PowerBuilder window object that is used to present and manipulate information in the application database without the developer having to code SQL statements and scripts. Other painters available are Picture Painter, to include bitmap graphics pictures in your application windows; Function Painter, to build your own PowerScript functions; Script Painter, for full-function text editing; and Preference Painter, to tailor PowerBuilder to your own PC configuration and personal preference. PowerBuilder also includes a fully integrated debugging facility that allows you to set any number of breakpoints and to single-step through the execution.

Conclusion

Client/Server ALLBASE is a product that provides a client/server programmatic interface to ALLBASE/SQL. It allows you to make full use of graphical user interface (like MS Windows 3.0) on the client for end-user presentation services and the excellent OLTP capabilities of the server to manage large amounts of data. You can rapidly port the existing Client/Server applications to ALLBASE/SQL or create your own applications using the leading tools like Gupta Technology's SQLWindows and Powersoft's PowerBuilder. More experienced programmers can create their own applications by writing in C directly to the Gupta SQLBase/C APIs that are supported by ALLBASE/SQL. The architecture of Client/Server ALLBASE allows these applications to simultaneously access other databases on other servers.

NOTE: At the time this paper was written, the product structure was still being finalized. Support for applications developed with the 4GL tools mentioned above will be available before support for the applications developed with C is available.

Acknowledgments

I wish to thank Graham Masters for his patient and useful technical guidance at innumerable times during the writing of this paper. I also wish to thank Adam Polson and my manager, Mariann Tymn, for introducing me to this product and being the reviewers of this paper.

References

1. SQLBase C Application Programming Interface Reference Manual, Gupta Technologies Inc.
2. SQLWindows Technical Summary, Gupta Technologies Inc.
3. SQLWindows — The Graphical Application Development Tool, Gupta Technologies Inc.
4. PowerBuilder — Product Overview, Version 1.0, PowerSoft Corp.
5. Microsoft Windows — Software Development Kit

Getting SQL Data Into NewWave

By ROBERT ROSS

Hewlett Packard
Software Technology Center
Roseville, California
916-786-8000
National INTEREX Conference: August, 1991

Why Do We Need This?

Why indeed would anyone want to take information out of "SQL" and place it into "NewWave"? Lots of people have lots of data stored in SQL databases that they would like to convert into easily understood information to pass along to others in the organization who make decisions based on what they know (or don't know). The NewWave environment is an ideal solution for combining different types of data into a compound document for printing, display, or electronic mailing to those who need to know. This paper is a case study of how one company combined both to provide solutions for their information flow problems, but before we get started, a few background facts might be helpful for some.

What is SQL?

The acronym SQL (Structured Query Language) has, through the influence of IBM and others in the marketplace, become synonymous with a type of database called **Relational**. However, the word "Relational" really describes how the data is stored, while "SQL" is a language that users and programmers alike use to specify how to retrieve that data. SQL language constructs could be used to get data from Network or Hierarchical databases, although this is rarely done. The SQL language and Relational databases pretty much go hand in hand.

What is NewWave?

The name "NewWave" has many different faces. You may have heard the terms *NewWave Desktop*, *NewWave Environment*, *NewWave Computing*, *NewWave Access*, *NewWave Mail*, *NewWave Write*, the *NewWave Development Environment*, and perhaps others. What I'll be talking about in this paper is the **"NewWave Environment"**, which is what you receive in the box when you purchase "NewWave" (version 3.0) from HP or a computer dealer. It has many parts, including the object-management and task language (Agents) software, and is installed on top of Microsoft Windows. It also includes the word processing system called "NewWave Write" which provides the capability to create what are called "Compound Documents". These documents may include many different types of objects, including text, graphs, spreadsheets, and pictures, all within a single object. The "originals" may even be located somewhere else, and when they are changed or modified, the version in the compound document will change also.

A Case Study: Let's Go Gambling!

To give you a feel for what we were trying to accomplish with this project, let's say you are the data processing manager of a Las Vegas casino. You have several large computers with databases that store information spanning payroll, general accounting, management reporting, hotel reservations, employee schedules, travel, casino floor cash inflow and outflow, etc. In this case it is that last category of "casino floor cash flow" that your boss is particularly interested in. He wants access to regular reports that show what's actually happening with his cash flow; plus daily, weekly, and monthly summaries.

To narrow the focus of the case study, we'll concentrate on the 3 most common types of gambling machines, which also happen to be the ones which are electronically tied to one of the large computer systems via a group of PC connections which collect real-time data on machine transactions at all times. These machines (Slot, Poker & Blackjack) create an "input" cash transaction record each time someone inserts some coins or tokens and plays the game through to completion, at which point another "output" cash transaction may be created if the player wins some money.

Each transaction consists of a record containing the System ID (the machine's assigned serial number), a time stamp, and a positive or negative amount. This data is collected in various transaction log files by the PCs, which at regular intervals interrupt an HP 3000 S/960 and cause some special software to read and process each PC log file, combining and writing the records into an ALLBASE transaction table. The software also updates a summary table which tracks each machine, its location, its current net "take" or balance, plus some other information.

What Does Your Boss Want To Know?

There are 3 main types of reports that your boss has asked you to have available on a regular basis, plus a few "ad-hoc" reports that may be requested at any time.

* Total Revenue for All Machines
* Revenue by Type of Machine
* Revenue by Location of Machine

You know from past experience that your boss likes to review revenue summaries each morning when he arrives. You also have learned that on busy weekends and holidays he may come running in at any time and demand to know what's going on in a particular room or location, so you need to be able to run off quick reports without much notice. A final thing that you'd like to address is the fact that if your boss has some particularly good news, he likes to write up a memo, include your numbers in it, and send it off to some "higher-ups" so they know what a good job he's doing.

Where's the Data?

The customer's data was stored in an ALLBASE database on an HP 3000 S/960. We were pulling our data from a table with some the following columns:

System_Id	Location	Balance	Trans_Lo	Trans_Hi	Total_In	Total_Out
SL1400A06	BL-A6-12	486.00	1.00	5.00	938.00	809.00
PO2250C15	BL-D1-04	160.00	1.00	5.00	455.00	330.00
BJ9000X24	BL-D1-19	1375.00	5.00	25.00	2670.00	2280.00
...	...	...	...	...	...	...

This data could just as easily have been stored in an ORACLE database on MPE/XL, or in an ALLBASE or ORACLE database on an HP9000 S/8xx, or even in a TurboIMAGE network database - the PC interface for the end user would be identical in each case.

What Data Do We Need?

To get the net revenue for any Slot, Poker, or Blackjack machine, we need to pull in the **System_Id** for identification, then subtract the **Total_Out** from the **Total_In**. Even though the database stores the machine's current balance, it needs to subtract out the value of the machine's initial balance, which is included in the current balance, but doesn't count towards net revenue.

How Are Going to Display This Data?

The "normal" way to generate a report listing this data would be to write several COBOL or 4-GL programs to read serially though the table, summarize the data according to type of machine, location, or group it as a whole. These programs would then be part of a job that could be streamed off on a regular basis or whenever requested. These reports might look something like the following:

Date: 02/27/96 Page: 1

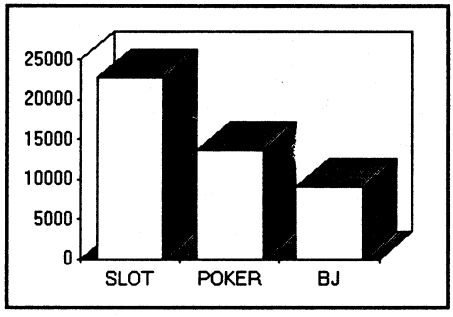
Electronic Slots Revenue Summary
Sorted by Machine Type
Summarized by Machine Type

SLOT	Blue Room	$16,430.00	-$15,040.00	$1,390.00
	Main Floor	$51,200.00	-$44,720.00	$6,480.00
	Masters	$108,250.00	-$93,320.00	$14,930.00
	. . .	. . .	. . .	. . .
	Totals:	$175,880.00	($153,080.00)	$22,800.00
POKER	Blue Room	$9,858.00	-$9,024.00	$834.00
	Main Floor	$30,720.00	-$26,832.00	$3,888.00
	Masters	$64,950.00	-$55,992.00	$8,958.00
	. . .	. . .	. . .	. . .
	Totals:	$105,528.00	($91,848.00)	$13,680.00
BLACKJACK	Blue Room	$6,572.00	-$6,016.00	$556.00
	Main Floor	$20,480.00	-$17,888.00	$2,592.00
	Masters	$43,300.00	-$37,328.00	$5,972.00
	. . .	. . .	. . .	. . .
	Totals:	$70,352.00	($61,232.00)	$9,120.00

What About a Graphic Display?

However, reports looking like this have been printed for decades - wouldn't it be nice to automatically convert these number into some type of graph or chart that would be easier to read - something that a manager could understand in a matter of seconds? Instead of showing your boss the normal report, what if you could show him something like this:

Electronic Slots Revenue Summary - 2/27/96



Admittedly, this is an oversimplification of the vast number of facts that any manager needs to know to made informed decisions, but there's no denying that a graphical display is a great deal easier to understand, in much less time, than a long (or even short) listing of numbers.

How Do We Get "Nice" Looking Data?

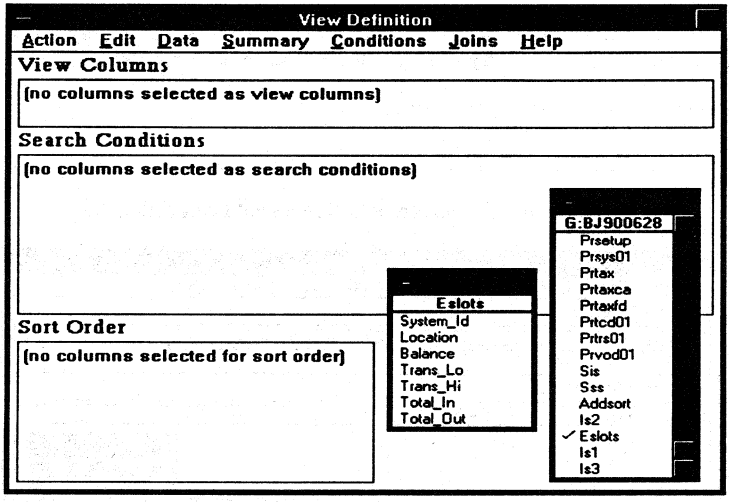
Graphical computer interfaces have a distinct advantage when it comes to presenting information using charts, graphs, or slides. Since all NewWave programs automatically come with a graphical interface, a program called **NewWave Access** makes an ideal solution for retrieving data from large SQL databases and displaying it in some sort of graphical format.

What is NewWave Access?

NewWave Access is a recently released product (Mar, 91) from Hewlett-Packard that integrates into the NewWave Environment and provides a simplified method for end-users to look at the data items that are available to them. They formulate a query by "dragging" and "dropping" the selected items in the right "box", requesting and viewing the data using an interface similar to Microsoft Excel, and then writing the results into various PC file formats.

When users first "click" on or open a NewWave Access object and get ready to retrieve some data, they are presented with a screen like the following:

NewWave Access Screen Example



There is really only 1 thing the user needs to do to get his or her data - let **NewWave Access** know which columns he or she wants to look at. The screen above shows 3 empty boxes (**View Columns**, **Search Conditions**, and **Sort Order**). You'll notice the larger window in the bottom-right of the screen (titled "G:BJ900628") - this is a list of all tables available to the user. By "clicking" on the entry "**Eslots**", another smaller window has been opened up to the left. This window contains all columns in that table that may be viewed by the user. Notice the small "checkmark" by the "**Eslots**" entry in the tables list - this indicates that table is currently being used.

In our case, what we want to do is pull out the following 4 columns from the **Eslots** table:

* SYSTEM_ID
* LOCATION
* TOTAL_IN
* TOTAL_OUT

If you're familiar with Microsoft Windows, you know that the cursor is usually in the shape of an arrow. To get our columns, we move the cursor into the "Eslots" window, where the cursor turns into the shape of a hand. When we "click" (press the mouse button) while the "hand" is pointing to one the column names, the hand sort of "grabs" it. If we hold the mouse button down and then move the cursor (hand), we can "drag and drop" the item into the top box labeled "**View Columns**". When we release the

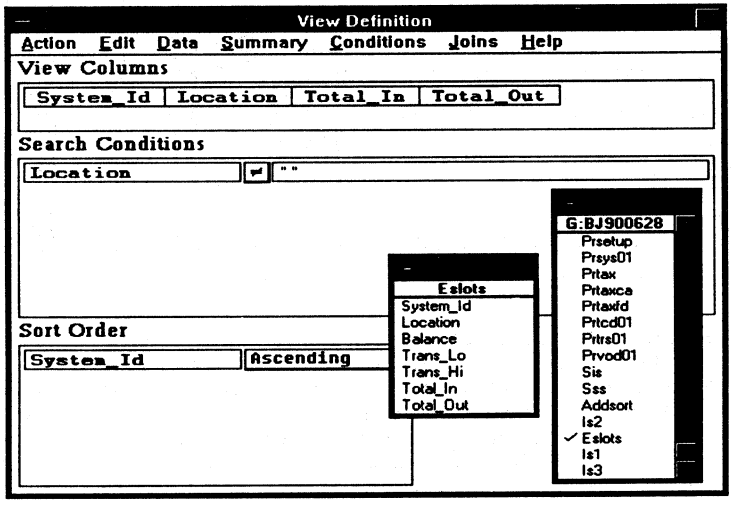
mouse button, the hand will "drop" the column name which will then appear as one of the "View Column" entries for the user to see. We can also "double-click" on a column name to accomplish the same thing. We do this for each of the 4 columns listed.

We also want to break down our revenue by machine type, so we need to sort on SYSTEM_ID, which includes machine type in the first 2 letters. Therefore, we also need to "click" on this column name to select it, then "drag and drop" it into the third "box" labeled "**Sort Order**".

Finally, just to see how it looks, we're going to add some criteria to the **Search Conditions** "box". We don't really need any criteria, so we'll just look for all **Locations** which are not blank.

When we're done, we have a screen that looks like this:

NewWave Access Screen with Requested Data Filled In



Once the user has filled in the required information in the **View Columns**, **Search Conditions**, and **Sort Order** "boxes", they click on the "**Data**" menu bar selection, and then choose "**Request Data**". NewWave Access will go out to wherever the data is stored and begin singling out the records from the database that match the criteria specified by the user in the **Search Conditions** "box".

In most cases, as soon as it finds enough records to fill up a screen, it will display them to the user while it continues to look for more records. The user can even be doing other work in the NewWave Environment, such as writing memos or working on spreadsheets, while the search continues in the background. The records are displayed to the user in a format that looks very similar to a Microsoft Excel spreadsheet, like the following example:

NewWave Access Data Display Screen

Eslots [NewWave Access]

Action Edit Data Connection Settings Task Help

BJ3684X01

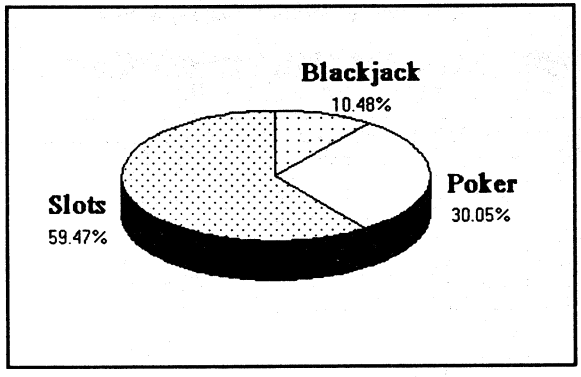
	System_Id	Location	Total_In	Total_Out
1	BJ3684X01	BL-D1-20	3022.00	2928.00
2	BJ3880X01	MA-B1-11	2428.00	2193.00
3	BJ3880X04	MA-B1-01	2241.00	1882.00
4	BJ3880X12	MA-BA-12	9430.00	8985.00
5	BJ8100X01	MA-B1-02	1421.00	923.00
6	BJ9000X24	BL-D1-19	2670.00	2280.00
7	PO2228C01	BL-D1-01	9834.00	7383.00
8	PO2228C02	BL-D1-02	3218.00	2199.00
9	PO2228C03	BL-D1-03	3258.00	2249.00
10	PO2250C15	BL-D1-04	455.00	330.00
11	PO6684C02	BL-D1-02	45685.00	42875.00
12	SL1400A01	MA-A1-01	85040.00	90090.00
13	SL1400A02	MA-A1-02	5840.00	4210.00
14	SL1400A06	BL-A6-04	938.00	809.00
15	SL1600A01	MF-LI-01	5250.00	4220.00
16	SL4456A07	BL-A6-01	9834.00	8989.00

Making Pictures Out of Data

Now that we have the data that we're looking for, we need to figure out what to do with it. Since the boss likes the "bar/pie chart" display, we can output this data into either a Lotus 1-2-3 or Microsoft Excel spreadsheet and use the graphing capabilities inherent in both to produce our bar and pie charts. Since the first release of NewWave Access does not output directly to the Microsoft Excel file format, we can output to a dBASE file format and then within Microsoft Excel read in the numbers directly from the dBASE formatted file.

To be able to display the total revenue by type of machine, we needed to be able to group on the first 2 letters of the SYSTEM_ID. These 2 letters were "SL" for Slots, "PO" for Poker, and "BJ" for Blackjack machines. The easiest way to do this was to create a separate 2-character column that was mapped to the first 2 letters of the SYSTEM_ID and group on this new column. We then sum up the net revenue for each machine type (Slots, Poker, & Blackjack), and end up with 3 numbers that will equal the net revenue for each type of machine for the whole casino.

The following is a sample pie chart of the results:



We can also change the view table to group and summarize on the location, substitute a room name, floor level, or building name for the generically coded location, and produce a chart that shows how much money is being made in each location. Or we could combine all machines and display a chart for the whole operation.

Can We Display These Graphs in our Reports?

Although your boss is very impressed by these graphs (and his are in color!), one thing that we mentioned at the beginning that we haven't done yet is to incorporate this information into a report that can be sent to "big wheels" to impress them when things are going right.

The **NewWave Write** word processing system that comes with the **NewWave Environment** makes it easy to create a compound document that starts with generic text and adds numeric or graphical information that gets automatically modified each time the report is run. Once the data is in a Microsoft Excel or Lotus 1-2-3 spreadsheet and a chart has been created, it may either be copied into the document through Charting Gallery, the Clipboard, or a graphics picture bit manipulator.

To get an idea of what this compound document can look like, here's a simple example:

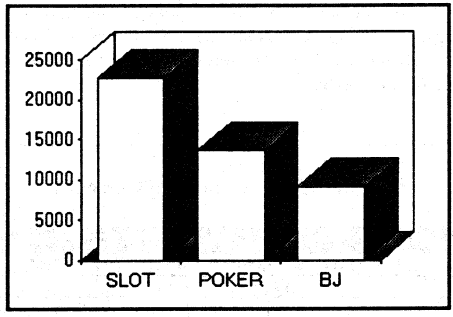
Hi-Li Casino
3897 Las Vegas Blvd.
Las Vegas, NV 89109

From: John Summalt
Data Processing Manager

Gentlemen,

Please note the following graph describing the performance of our latest "D" models of Bally's slot machines, video poker players, and blackjack machines.

You'll notice that at current levels, the break-even point is set for 8 months, rather than the usual 12-14 months. We're very pleased with these results, and I'd like to thank you for making the decision to purchase these models.



Sincerely,

JS:mb

What Have We Done So Far?

Our customer has now gone through a process of using tools and creating procedures that will help him or her in doing the following:

* Find out what data is available to be looked at
* Use a simple "fill-in-the-box" method of selecting what information to view
* Let the system retrieve data in the background, freeing the user to do other work
* Display and refine the data until it looks the way the user wants
* Save the data in a PC file format for storage, collection, or further manipulation
* Be able to save the "Data View" definition for next time

Even though users really enjoy the simplicity of this process, whenever there are repetitive tasks to be done daily, week, or monthly, the ability to automate them so all the user has to do is "push a button" and get the finished result can really take the pressure off of someone and free them up to do other jobs.

The NewWave Environment can do this type of automation using **"Task Menus"** or **"Agents"**. These both use a task language along with a "recorder" that can be turned on while the user executes an operation. Every time the user types something, selects a menu choice, enters a different window, etc., the "recorder" registers that action or keystroke and writes a sequence of commands to a file using this "task language". When the user is finished and turns the "recorder" off, an **"Agent Task"** has been created.

These tasks may be placed into a menu structure, or they may be objects on the user's NewWave Desktop that can be "dragged and dropped" on the **"Agent"** for execution. They may also be scheduled for multiple executions at certain times of the day, week, or month (if you don't mind leaving your computer on) by using the Agent calendar.

Sample Agent Scheduling Calendar

Agent Calendar

Action Edit Objects Days Dates Task Help

February 1996

Sun	Mon	Tue	Wed	Thu	Fri	Sat
Repetitive Schedule						
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

So How Do We Automate The Process?

Wouldn't it be great if we (the computer literate "we") could take a process such as the one described above - retrieve some data, format it into a graph, and paste it into a professional looking report for management - and automate it to the degree that we could place a file on our manager's computer, show him or her which button to push (or object to click on), tell them to take a coffee break, and when they return the finished product will be ready for them to review? With a little trial and error (done on the privacy of your PC!) and using the tools and products already mentioned, this can easily be done.

When a user purchases and installs **NewWave Access**, one thing they will notice on their NewWave Desktop is a new folder called "*NewWave Access - Ideas*". This folder contains several "ready-to-run" tasks that may be used as examples of how to build your own Agent or Menu tasks. Although each situation will be different, many of the concepts and ideas presented in this folder and the "how-to" will still apply when you're trying to create your own automated task.

Which 20 Machines Make The Most Money?

Our customer mentioned repeatedly that his boss always wants to know which machines are making the most money so he can figure out why. We suggested that he implement a task that will display a list of the top 20 machines in "money-making" order and copy it onto the PC his boss uses so that he could get the list any time he wanted. This procedure could also apply in any circumstance where someone wants to know who his top salespeople are, or which of her product lines are selling most quickly, or which inventory items have the greatest backlog.

How Did We Get It Done?

A. Within the NewWave Environment, we first created a NewWave Access object. In our database we specified a new view that included an additional column, **Net_Rev.**

Net_Rev = Begin_Bal + (Total_In - Total_Out)

B. We then opened the new table and pulled out **System_Id, Location,** and **Net_Rev** into our NewWave Access view definition. We sorted the rows in descending order based on **Net_Rev.** Finally, we saved this data view.

C. Next, we recorded an Agent Task while doing the following:

- Update the NewWave Access object with current data from the database
- Find the value of **Net_Rev** in the 20th row
- Define a second view definition where: **Net_Rev > (value** in 20th row)
- Convert this view into a PC spreadsheet format, or
- Copy and Paste these rows into an already formatted report

D. Finally, we turned off the Agent Task recorder to save the task for later repeated use.

What Did The Task Look Like?

While an Agent Task is being created, lines of the Agent Task Language are being written to a file that may later be modified, simplified, improved, or commented. To get an idea of how this looks, here are a few samples of the language.

The first example shows how to select the table, get the columns to display, make sure they're sorted in the right order, and request the data to be retrieved. For this example, we added a search criteria, although it wasn't needed in the actual view.

Agent Task Language Example - Define & Get Data

```
PROCEDURE DEFINE_VIEW
 '*
 '* Select the table by name (eslots.revenue)
 '*
  FOCUS NWACCESS nwaccess_obj_name#
  SHOW_TABLES_LIST
  SELECT_TABLE "ESLOTS.REVENUE"
  SHOW_COLUMNS_LIST
 '*
 '* Pick the columns that we want to look at
 '*
  SELECT_COLUMN "SYSTEM_ID"
  ADD_TO VIEW_COLUMN 1
  SELECT_COLUMN "LOCATION"
  ADD_TO VIEW_COLUMN 2
  SELECT_COLUMN "NET_REV"
  ADD_TO VIEW_COLUMN 3
 '*
 '* Put in any search criteria (location <> "")
 '*
  ACTIVATE COLUMNS_LIST  "ESLOTS.REVENUE"
  SELECT_COLUMN "LOCATION"
  ADD_TO SEARCH_CONDITION 1
  ACTIVATE VIEW_DEFINITION
  SELECT SEARCH_CONDITION 1
  SET_VALUE " "
  SET_OPERATOR NOT_EQUAL
 '*
 '* Add in sort order (sort on net_rev)
 '*
  ACTIVATE COLUMNS_LIST "ESLOTS.REVENUE"
  SELECT_COLUMN "NET_REV"
  ADD_TO SORT_ORDER 1
  ACTIVATE VIEW_DEFINITION
  SELECT SORT_ORDER 1
  SET_DIRECTION DESCENDING
  RETURN
ENDPROC
-------------------------------------------------------------

PROCEDURE REQUEST_DATA
 '*
 '* We're now going to ask NewWave Access
 '* to retrieve our data for us
 '*
  FOCUS NWACCESS nwaccess_obj_name#
  ACTIVATE DATA_VIEW
  REQUEST_DATA
```

The next part of the example contains the "critical" part - getting the value of **Net_Rev** that gets displayed in the 20th row and then using that value as the search condition for the next data request. The end of the procedure also selects all of the data and copies it into the Clipboard to be used later.

Agent Task Language Example - Only Get Top 20

```
PROCEDURE GET_TOP_VALUE
  top_num#=GET_CELL ("NET_REV",20)
  top_num#=STR(top_num#)
  SHOW_VIEW_DEFINITION
  ACTIVATE COLUMNS_LIST  "ESLOTS.REVENUE"
  SELECT_COLUMN "NET_REV"
  ADD_TO SEARCH_CONDITION 2
  ACTIVATE VIEW_DEFINITION
  SELECT SEARCH_CONDITION 2
  SET_VALUE top_num#
  SET_OPERATOR GREATER_THAN_OR_EQUAL
  REQUEST_DATA
  SELECT_ALL
  COPY
  CLOSE SAVE
  RETURN
ENDPROC
```

The final part of the example selects a NewWave Write object that we're going to paste/share our data values into. This document already has text in it - we're just adding a list of numbers to support or demonstrate our conclusions or statements. The "ADD_TAB_STOP" statements help to make the data columns line up for easier reading.

Agent Task Language - Paste Top 20 Data Into Document

```
PROCEDURE PUT_DATA-IN-DOCUMENT
  FOCUS FOLDER folder_name#
  OPEN
  FOCUS NEWWAVE_WRITE nwwrite_obj_name#
  !*
  !* Add tab stops to make data easier to read
  !*
  PARAGRAPH TABS
  ADD_TAB_STOP AT 1.5 INCHES LEFT
  ADD_TAB_STOP AT 3.0 INCHES LEFT
  ADD_TAB_STOP AT 5.0 INCHES LEFT
  NEW_PARAGRAPH 2
  PASTE
  RETURN
ENDPROC
```

What About The Long Haul?

Once our customer's boss was able to find out which machines were making the most money each day, suddenly he wanted to see what was happening over a period of time, rather than a daily "snapshot".

We took the previous task to display a list of the top 20 machines in "money-making" order and added new instructions to save these daily "snapshots", so at the end of the month (or anytime) we could graph what was happening over time. This procedure could also apply to any circumstance where someone wants to see "target vs. actual", or "ordered vs. shipped", or needs to identify trends before they get out of hand.

What's Different About This Task?

A. Within the NewWave Environment, we first created a NewWave Access object. We used the same view as last time with the new column, **Net_Rev.**

B. Since the database contains data over a long period of time, we needed to add a data parameter to be compared against the transaction timestamp so that if we run our task once per day, we'll only pull out that day's transactions. We don't need to sort on **Net_Rev** this time - we'll sort it later for the report.

C. Next, we recorded another Agent Task while doing the following:

* Update the NewWave Access object with current data from the database
* Find the value of **Net_Rev** in the 20th row
* Define a second view definition where: **Net_Rev** > (**value** in 20th row)
* Convert this view into a PC format that supports appending, such as dBASEIV

D. Finally, we turned off the Agent Task recorder to save the task for later repeated use.

E. Each day when the Task was run, we asked the user to confirm today's date as the time period of collection, then appended the data to the same local table each day, so at any time this table would contain all of the data for the month.

What Were Some of the Changes Needed for This Task?

This example shows how to convert the data requested to a dBASE file. We used a local dBASE file because it supports the "append" option. The first time that we output the data each month, the file needs to be written in "create" mode - all other times it needs to be written in "append" mode.

Agent Task Language - Convert Out in Create or Append Mode

```
PROCEDURE GET_AND_APPEND_DATA
  WHILE count# < day#
    IF count# <> 1
      SHOW_VIEW_DEFINITION
      ACTIVATE VIEW_DEFINITION
    ENDIF
    SELECT SEARCH_CONDITION 3
    date# = STR(year#)+STR(month#)+STR(day#)
    SET_VALUE date#
    REQUEST_DATA
    IF count# =1
      CONVERT_OUT TO LOCAL_DBASE4_TABLE
     "ESLOT_REVCUM" ON CONNECTION "Local
      Database" CREATE
    ELSE
      CONVERT_OUT TO LOCAL_DBASE4_TABLE
```

```
    "ESLOT_REVCUM" ON CONNECTION "Local
     Database" APPEND
   ENDIF
   count#=count#+1
  ENDWHILE
  RETURN
ENDPROC
```

Problems?

One problem we ran into was the fact that every day a different set of machines would be in the Top Twenty. When it came time to combine them all together at the end of the month, a few machines would be in the file every day of the month, but most of them didn't end up in the file on a daily basis. What we decided to do was modify the original task to sort and group (summarize) by machine type and location. This allowed us to have a record in the file every day for each type of machine in each room, making it much easier to have all of the data points available for graphing at the end of the month.

The next part of the example shows one of several tasks done at the end of each month to query the local table, pick up all records belonging to either a machine type or a location, and save them in a separate dBASE file so we can go back and produce a series of graphs from them later.

In this example, the Search Conditions are set up for Machine-Type = "BJ", so this will be the file to produce a graph for all Blackjack machines.

Agent Task Language Example - Monthend Data Extraction

```
PROCEDURE GET_ALL_BLACKJACK_RECS
  FOCUS NWACCESS nwaccess_obj_nameB#
  SHOW_TABLES_LIST
  SELECT_TABLE "dBASE.ESLOTS_REVCUM"
  SHOW_COLUMNS_LIST
 '*
 '* Pull out only the columns needed for producing the graph
 '*
  SELECT_COLUMN "MACHINE_TYPE"
  ADD_TO VIEW_COLUMN 1
  SELECT_COLUMN "LOCATION_GEN"
  ADD_TO VIEW_COLUMN 2
  SELECT_COLUMN "REC_DATE"
  ADD_TO VIEW_COLUMN 3
  SELECT_COLUMN "NET_REV"
  ADD_TO VIEW_COLUMN 4
  SELECT_COLUMN "MACHINE_TYPE"
  ADD_TO SEARCH_CONDITION 1
  ACTIVATE VIEW_DEFINITION
  SELECT SEARCH_CONDITION 1
  SET_VALUE "BJ"
  REQUEST_DATA
      CONVERT_OUT TO LOCAL_DBASE4_TABLE
     "ESLOT_REVCUMBJ" ON CONNECTION "Local
      Database" CREATE
  RETURN
ENDPROC
```

Once the monthly file has been separated into Slot, Poker, and Blackjack files, plus separate files for each general location, we can use the same algorithm on each one (with different titles) to produce line charts from each file using either Microsoft Excel or Lotus 1-2-3. These charts are then copied through the Clipboard into a Compound Document, which might look like the following:

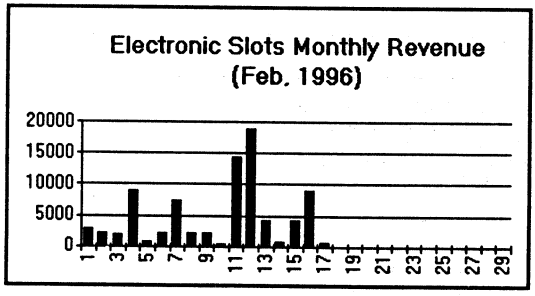
Compound Document with Current Revenue Totals Graph

Hi-Li Casino
3897 Las Vegas Blvd.
Las Vegas, NV 89109

From: John Summalt
Data Processing Manager

Bob,

I thought you might be interested and pleased by the following results so far for February. Our overall results are not only way ahead of last year this time, but significantly ahead of our projections for our fiscal year-to-date totals.



Sincerely,

JS:mb

These types of memos, once created and stored as a NewWave object, can be automatically updated with new data every night so a manager always has current information on his or her desk every

morning. If you really want to save paper, this type of information can be sent via electronic mail, or displayed in a "slide show" format with each graph representing different measurable statistics that managers use to gauge their business.

This data could also be sent to a manager as a spreadsheet if they like to sit down and play with the numbers to try and come up with forecasts or other figures.

What Did We Conclude?

Once end users (in this case, managers) find that they can get their hands on all sorts of data that wasn't easily available before, they have a tendency to "go wild". We found that pretty soon they wanted all types of bar graphs, line and pie charts, etc. We also found that they are prone to making a big deal out of one, but as soon as they get another one, they forget all about the work you put into creating the first one.

When the "final product" is ready to go and copied over onto the manager's PC, all they had to do was "push a button" to get what they wanted - they didn't realize that a lot of thought, design, and testing went into making it so very simple for them. We finally had to sit down and map out exactly what metrics they were going to use to measure the progress of their business (or lack of it) and produce some charts to demonstrate those numbers.

Nevertheless, the concept of pulling collected data from a large database into your PC, massaging it, producing graphs or charts from it, and finally including it as part of a memo - and especially having all of this automated and/or scheduled for you - was proven over and over again as new managers became excited about getting "their data" in the same manner. They liked the end product, and we had fun learning how to produce it for them.

MANAGING PC S/W: A BETTER WAY

Ron Slone
Hewlett Packard
8000 Foothills Blvd
Roseville, CA 95678
(916) 786-8000

As PCs continue to abound throughout the business environment, the cost in dollars and time to manage the PC software distribution function continues to climb. MIS groups continually struggle with ways to get a better handle on their installed base of PCs and simultaneously contain the costs of delivering PC support services.

This paper presents a strategy for a centrally controlled, cost-effective and automated PC software management system with a focus on a HP-developed approach to the problem. Issues such as cost justification, software installation/removal, licensing control, version control, PC audits, and hardware tracking will be addressed. It will illustrate how PC software and hardware can be more closely managed at a lower cost with no reduction in support services.

SCENARIO:

Picture, if you will, a corporate headquarters building. Inside the walls of this build lurks the following PC environment:

- 200 networked PCs operated by employees with little PC savvy.
- A variety of PC brands, processors, video display units, and printers.
- Five or more applications are installed on each PC. There are multiple versions of some of the applications.
- The environment is very labor intensive for the MIS department due to changing of PC configurations and installing software.
- Many end users have installed software "borrowed" from other users.
- Hardware and software configurations information is available on bits and scraps of paper in the Technical Support department.

You may recognize some of these problems in your own company. PC management problems typically manifest themselves at PC audit time each year. Passing an internal audit tends to become very frustrating and time consuming.

This presentation gives a technical perspective of a tool named HP Software Vendor. After a technical introduction, we will apply the use of the tool to resolve some of the problems noted previously.

HP Software Vendor is a PC-level application used to track PC hardware, install software over the network, track available and installed software, and track the license counts for the installed software. I will discuss using HP Software Vendor to install PC applications, but any PC files can be distributed using the tool.

Two important objectives of HP Software Vendor are to centralize the PC management and tracking process and to provide a means to track software licenses to avoid legal legal issues with software providers.

The Software Vendor components reside in a network directory, a system administrator's PC and on the end user PCs. All PCs must have access to a common network server.

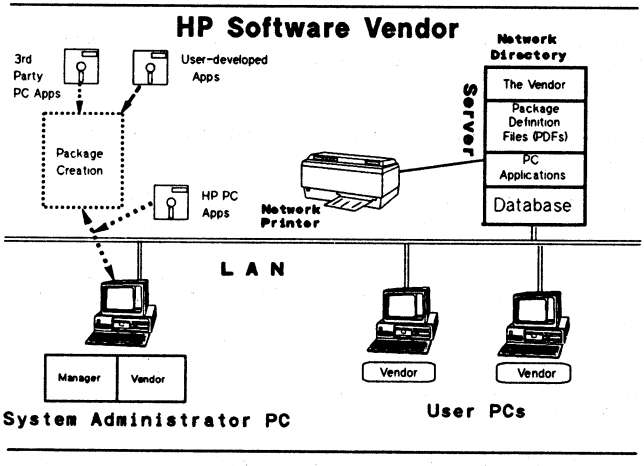


Figure 1. HP Software Vendor Network Environment

HP Software Vendor requires Microsoft Windows 3.0 on the system administrator PC. End user PCs require MS-DOS 3.1 to 4.01.

HP Software Vendor introduces the concept of "packaging" applications in order to install them to end user PCs. A package consists of the application to be installed; a database containing such items as the name and location of the software to be installed; and a package definition file (PDF) which is a list of commands that perform the installation. Let's look at each piece of the package in more detail.

Application to install: the application resides in a network directory accessible to the end user PC and the system administrator PC.

Database: A collection of ten databases form the core of HP Software Vendor. The databases contain the following data items:

1. PC database contains PC name, password, configuration values.

2. Packages stores package names, package connection path, package definition file name.

3. License groups DB stores names of the application group for which license tracking is to be done.

4. Status DB maintains status of packages, shows if package is allocated or installed, installed PC id, and package id.

5. Values DB contains names of the different PC components that can have different configurations. Some of the values are the "PC Model", "Display", "Video Card".

6. PC configuration values for the names in the Values DB. For example, under "PC Model", the default list contains "Vectra 286", "Vectra 386" et al. If the correct PC model for the end user is in the list, it is selected. Otherwise, the system administrator can add the PC models used in the corporation.

7. Package creation errors DB stores errors generated when creating package definition files.

8. PC notes text DB stores additional information about end user PCs in addition to the configuration data.

9. Package notes text DB provides a communication area for the system administrator to use to communicate with the end user at the time the package is installed.

10. Installation errors DB holds errors that occur during installation to the end user PCs. The system administrator can review or print the errors for correction.

Package definition file: a list of high-level commands perform installation of the software from the network directory to the end user PCs. This language has a rich set of commands that also can perform many of the configuration requirements needed on the end user PC.

HP Software Vendor is divided into two major functional areas: the system administrator function and the end user function. The system administrator performs these activities:

- installs the application to the network directory
- enters end user PC data into database
- develops the package definition file to do the installation to the end user PCs
- assigns the package to all PCs to be installed
- sets up license tracking if required for the application

The primary end user function is to initiate the physical installation once a package has been assigned to his or her PC by the system administrator.

Let's now discuss how HP Software Vendor resolves some of the problems listed in our scenario.

PC Tracking

The capability to track PCs and their hardware configurations is available to the system administrator.

This capability can be done independently of software installation, though the PCs must be identified in the database before software can be installed to them. Currently, the system administrator has to perform a manual collection of data on the corporate PC installed base. As much or as little information as desired may be entered into the PC database. Figure 2 illustrates the screen for entry of PCs and hardware configurations.

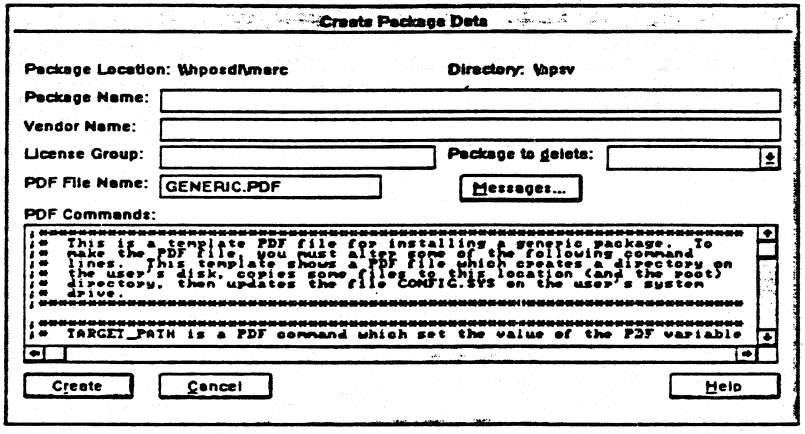


Figure 2. Add PC Data

Software Installation

The system administrator uses the "Create Package Data" screen to create the package information and develop the package definition file to install the software. The software is may also be assigned to a license group for license tracking. Figure 3 shows the screen used to enter the package information.

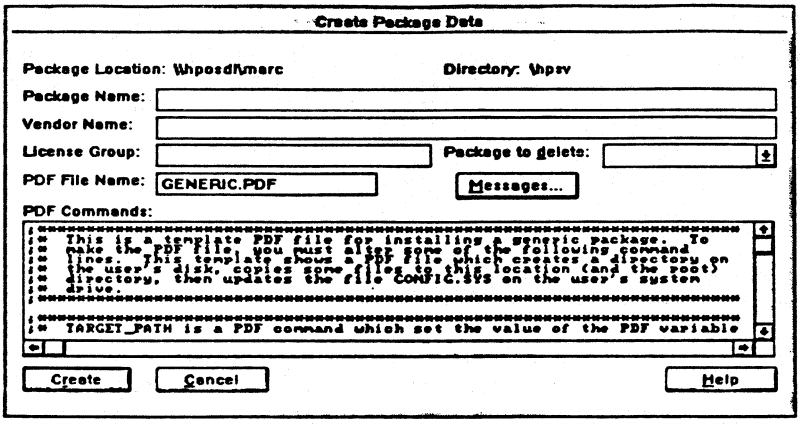


Figure 3. Create Package Data Screen

The removal of software from end user PCs can be accomplished by entering the package name to be deleted from the PC. Once the package has been created, the system administrator then uses the "Add PC Data" screen to assign the package to all end user PCs that will be installed with this package.

Since a support person doesn't have to go to each PC to assist in the software installation, the entire end user community can be on the same software version within a about a 24 hour period.

In addition, the overall support time is reduced since the software is prepared and installed from a central location.

A sample of a package definition file is shown below. The example illustrates the commands needed to install WordPerfect 5.0.

```
;The installation will not interact with the screen
NONINTERACTIVE
;Do not create the subdirectory if it already exists
NODUP
;Accept upper or lower case commands
NOCASE
;
;Direct the WordPerfect 5.0 installation to C:\WP50
TARGET_PATH C:\WP50
;
;Create the \WP50 subdirectory on the hard disk if it
;
MD %TARGET_PATH%
;copy files in \WP50 to the designated drive and
;subdirectory
COPY %SOURCE_DRIVE%%SOURCE_DIR%WP50\*.* %TARGET_PATH%
;
;NOTE: A separate PDF can be created to vend LEARN
;WordPerfect to selected user PCs if desired
;
;By default add the following as the last line in
;CONFIG.SYS
MODVALUE %SYSTEM_DRIVE%\CONFIG.SYS FILES 20 >
MODVALUE %SYSTEM_DRIVE%\CONFIG.SYS BUFFERS 5 >
;
;Add path to the PATH string in AUTOEXEC.BAT, otherwise
;create a PATH string
ADDPATH %SYSTEM_DRIVE%\AUTOEXEC.BAT PATH %TARGET_PATH% END
;
;End of WordPerfect 5.0 installation
```

Figure 4. Package Definition File for WordPerfect 5.0

The end user runs the Vendor component of HP Software Vendor to install the package from the network directory to their hard disk. The packages to be installed are selected from those assigned, but not yet installed. Figure 5 depicts the interface on the end user PC.

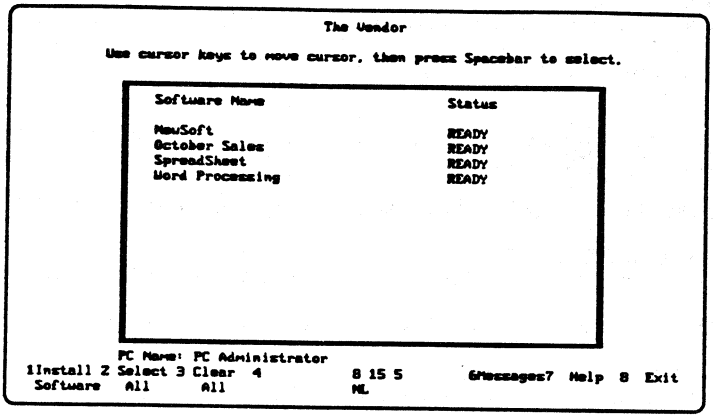


Figure 5. End User Interface

In our example, the end user would select "Word Processing" as the package to install. After selecting all the packages to be installed, the user presses a function key to trigger the actual installation. After a successful installation, the selected software is tagged with "DONE" to show it has been installed.

In addition to installing the software, our package definition file also modifies the CONFIG.SYS and AUTOEXEC.BAT files on the end user PC after the package has been installed. These charts show the changes that were made:

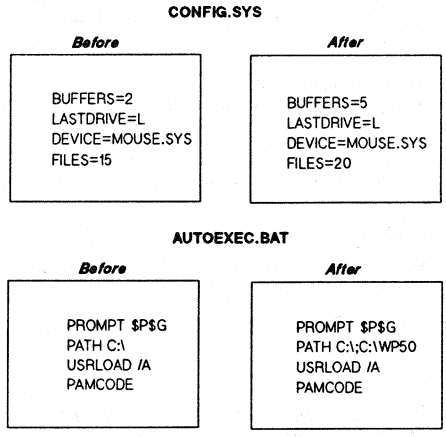


Figure 6. Modifications to CONFIG.SYS and AUTOEXEC.BAT

License Tracking

With the advent of site licensing or right-to-use licenses, there is not a need to buy the software in individual packages. A small number of reference manuals can be strategically placed in case they are needed by the users.

The system administrator sets the number of licenses available for an application. The installation process adds to the count of licenses used. When all available licenses are used for a license group, HP Software Vendor generates an error message and does not allow additional installations until the available count has been increased for that license group. If a licensed application is removed from an end user PC, the license count is also adjusted and that license is available for another user.

This approach minimizes the opportunity for unauthorized installations of software. For non-licensed software, the count fields can be used to track the number of applications installed.

The following screen is used to set the license counts.

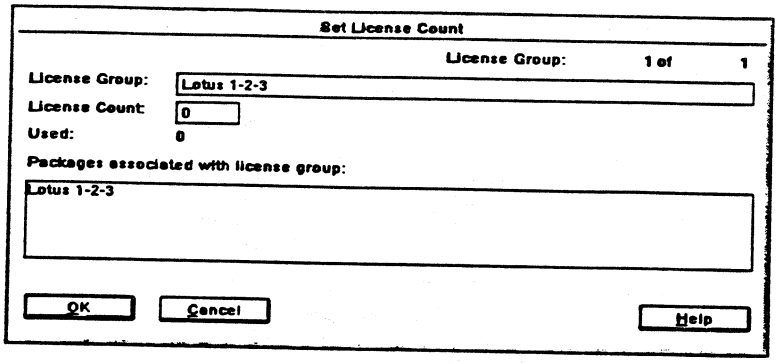


Figure 7. Set License Count

Reporting

The system administrator may select up to four reports from the database; PCs, Packages, License Groups, and Errors. The errors report is a listing of errors generated during the software installation process. The reports may be printed or exported for use in spreadsheets or other databases.

Cost Justification

This tool provides convenience, greater control of your PC environment and software tracking capabilities. More tangible benefits are the reduction of time and effort. This is a rule of thumb guideline to help you cost justify the investment in a software management tool.

1. Determine the number of applications that will be installed per annum.

2. Estimate the number of updates each application will have each year.

3. Determine the approximate amount of time needed to currently install/update each application.

4. Determine the hourly value of your end users time

Disregarding the cost of the technical support effort, the formula for your estimated cost savings is:

Number of installations/updates x time to install/update x hourly rate = cost savings per end user

PAPER NUMBER: **4901**

PAPER TITLE: **HOW TO INVENTORY CITY TREES WITH PARADOX & GIS MAP**

AUTHOR: **Augusta C.N. Crutchfield**

ADDRESS: **City of Sunnyvale -- IMS Dept.**
650 West Olive Avenue
P.O. Box 3707,
Sunnyvale, CA. 94088-3707, USA.

TELEPHONE: **(408) 730-7657**

[4901.A] BACKGROUND:

City of Sunnyvale was incorporated in 1912 and became a charter city in 1950. It operates under a council-manager form of government. The seven-member City Council appoints a City Manager who appoints all other employees according to provisions of the charter and the Civil Service Rules and Regulations. Sunnyvale is an industrial/residential community located 40 miles south of San Francisco with an employment base of approximately 135,000. Sunnyvale is generally considered the "heart of Silicon Valley" with high-tech industries comprising the majority of Sunnyvale's industrial sector. Education levels tend to be high with many professionals in Sunnyvale's residential base.

Sunnyvale's current population is 117,000 within the 24 square miles of the City. Build-out population is estimated to be 130,000. Approximately 24% of the population are under 18 years of age, approximately 18% are over 65, and approximately 25% are minorities.

Sunnyvale's city government has historically been considered very progressive. The primary issues facing the City Council and staff have been and will continue to be transportation, open space, housing, and continued financial stability. The total 89/90 budget is $115 million. The operating budget is $74 million, capital budget and equipment purchases is $37 million, and $4 million is debt service. The City has approximately 700+ full-time employees plus about 600 seasonal workers spread over the following operating departments, i.e.

* Office of the City Manager
* Office of the City Attorney
* Public Safety
* Employment Development
* Community Development
* Parks & Recreation
* Information Management
* Public Works
* Libraries
* Finance
* Human Resources

The Street Trees and Landscape Division of the Department of Public Works in the City of Sunnyvale, is responsible for the maintenance and inventory of all City-owned trees and the replacement and maintenance of the median strip landscape and concrete sidewalk. There are approximately 33,000 City-owned trees that need structural pruning every four years. The City has mandated the Division to identify, track and replace all incidence of sidewalk damage caused by tree roots and related tree damage.

[4901.B] OBJECTIVES:

The Street Trees and Landscape Division has a high level of visibility with Sunnyvale residents. One of the joint City and Citizen goals is to continue high standards for installation and maintenance in landscape requirements, including preservation of existing trees where possible. The main selected service level objective in the Division is to perform structural pruning of all trees by the fiscal year 1988/89 and provide block-by-block maintenance pruning to 33,000 City-owned trees every 4 years at the rate of 25% of the total street tree count each year. Another service level objective is to provide tree services that require priority work within 3 weeks of request & complete other non-priority work within 6 weeks, 90% of the time.

[4901.C] CUSTOMER REQUIREMENTS:

The Information Management Services (IMS) division of the City of Sunnyvale launched off a project to assist its customers, the Street Trees and Landscape Division, to design and develop a brand new automated and user-friendly Street Trees Inventory system that utilizes to some degree the current technology within a limited budget constraint. The main customer requirements are:

a. Need capability to assess the trees information in the

database more easily with user-friendly features and controls of entering/updating citizen requests into the system.

b. Need to have more timely response to citizen requests and emergencies, e.g. tree trimming requests and potential danger of dangling branches, uproots, etc.

c. Need to have better organization and scheduling of work crews to very specific tree sites and tree locations.

d. Need to have detailed output reports on progress and status of specific contract firms on tasks, e.g. tree trimming work.

e. Need an up-to-date automated tree inventory and management system for more hands-on & accurate analysis/ planning of potential tree removal and replacement of the maturing stock of trees in a more efficient way.

f. Need some portability features of performing instant data entry onto the new proposed TreeBase database and software application.

g. Need system integration features with another modular system, such as the concrete sub-system, with the proposed TreeBase system, in terms of evaluating sidewalk and curb damage that is caused by tree roots and other foliage growth; plus scheduling/projecting potential damage to sidewalks, etc.

[4901.D] PROCEDURES OF THE CURRENT TREEKEEPER SYSTEM:

The City Arborist enters the tree information into the current TreeKeeper system from a vendor, and generates a work request. The arborist will then evaluate the work request to determine the type of work to be done. When a decision is made to perform the work, the arborist generates a work order. The work order is assigned to a crew or an inspector, depending on the action noted on the work order form, which is generated by the TreeKeeper system. Once the work is completed, the form is returned to the arborist to update the work order data into the TreeKeeper system. Generating scheduled or predictive maintenance work orders on a weekly and monthly basis for activities such as tree trim, root pruning and miscellaneous tree maintenance (e.g. watering, fertilization, etc.) is another required task. Work order forms are sorted and dispatched to the crew leaders each week. Upon completion, the work orders are updated by the arborist, who then generates reports on a weekly, monthly and

quarterly basis, for the end-users at the Street Trees and Landscape Division.

The Division generates 10 to 20 emergency work orders and 30 to 40 scheduled work orders per week. Emergency work orders usually reflect one or two tree sites per work order. A scheduled work order can consist about 30 to 200 trees and sites for scheduled maintenance and review. Emergency work orders are completed within 24 hours of a call.

The need for accuracy and quality is very important. The Street Trees staff take great effort in managing both the manual and automated sub-systems with about 95% accuracy. The main accuracy problem originates from the inaccuracies of the current City TreeKeeper database at the time of data collection. Considerable time and efforts have been devoted to updating & correcting the database. Approximately 12 man-hours per day is devoted to maintaining the current City TreeKeeper database and processing work orders.

[4901.E] PROPOSED NEW TREEBASE SOFTWARE APPLICATION:

The main system objective of the proposed new TreeBase software application is to create a data collection and management system that will do the following:

- reduce the amount of paper flow
- simplify the flow of information within the Division
- meet the information requirements of the City
- enhance the service level to the Sunnyvale community
- provide greater end-user control and flexibility

The new TreeBase system will be a prototype of an inventory and maintenance application system that utilizes Borland's PARADOX 3.5 relational data base system in a multi-/single-user environment; and customized programming modules that are written and developed in C, an object-oriented programming language, with graphical user interface (GUI) on a City of Sunnyvale geographical information system (GIS) map template.

[4901.F] FEATURES & FUNCTIONS OF NEW TREEBASE SYSTEM:

a. A functioning, integrated, modular and "bug-free" system.

b. A system that can be shared/linked by all staff members within the Division.

c. A tree trim cycle sub-system that can be built-in to the maintenance system.

d. A concrete/sidewalk tracking & maintenance work order system that can be built-in linked to the main Treebase inventory system.

e. A flexible, user-friendly and graphical user interface system that integrates input and output modules with the main TreeBase software application.

f. An automated historical work order system to convert and replace the existing manual system of 17,000 work records stored in 3x5 inched indexed card system.

g. A built-in system with enhanced modules to integrate future sub-systems of the department of Public Works,e.g. water-meter, sewers, traffic signs, etc., and has the potential interfaces with portable personal computers for field data entry, data collection and graphical mapping.

[4901.G] INFORMATION REQUIREMENTS

The main data information requirement is based on addresses -- the physical location of an entity. The primary entity is a street tree site. The site is qualified by:

- the type of an existing tree being planted
- physical characteristics and layout
- history and the effects on the site of the trees

All information tracking for all sections of the Street Trees and Concrete sub-systems revolve around these elements.

The type of tree is its species, e.g. the Magnolia tree. Its physical characteristics are elements, such as the height of the tree, and the diameter of the tree trunk.

The physical layout and history is the size of and shape of the plant well and its position that is relative to the street and curb.
The site history includes elements, e.g. planting date, previous plantings and all corrective and scheduled work done at that site and to the particular tree.

The effect of a particular tree on the location includes the relationship to elements, e.g. sewer line, power line and physical proximity to the structures.

[4901.H] OVERVIEW OF TREEBASE INVENTORY/MAINTENANCE SYSTEM:

The Information Management Services (IMS) Division of the City of Sunnyvale, has contracted with ATOPS Consulting to rewrite the existing TreeKeeper software in PARADOX 3.5 and modify the design by implementing a geographical information system (GIS) module. The IMS Division was able to implement this innovative PARADOX/GIS system within budget, via a contractual agreement to be a beta test site with ATOPS Consulting.

The GIS is a digital map that interfaces with PARADOX 3.5 TreeBase database. The provision of the digital map data is a complex task compared with the provision of conventional hardcopy map products. The compilation, editing, interfacing are more involved than the comparable phases of conventional map production. Furthermore, the application of a digital map data is far wider and extensive than the applications for conventional map.

Geographical information is identified by X,Y coordinates which can provide accurate locations of trees, signs, lights, sidewalks, water-meters, etc. The GIS module(s) would generate a map with selected information, such as trees and trim cycles. The user would be able to look at a GIS map and at any point on the map, would find information about a specific item and instantly bring up data relating to a particular item.

[4901.I] PROCESS DESCRIPTION OF TREEBASE:

The TreeBase program will reside on multiple personal computers and the TreeBase data will reside on a file server. In this way, multiple people can utilize the data simultaneously. The arborist will use a portable personal computer on the field. When he needs to update the information, he will be able to directly enter the data to the TreeBase database, via the portable personal computer. When the arborist returns to his office, he will simply download the updated information from the portable personal computer to the file server. The download process is automatic and less time-consuming. For other end-users who need to access the TreeBase information from another building, they can simply use a modem to push/pull data to and from the file server, where the master TreeBase resides. This is an interim and economic solution, whilst awaiting Phase II of the project to be approved for PC LAN integration.

[4901.J] TREEBASE SOFTWARE DESIGN:

The TreeBase system will provide the following GUI in entering tree information utilizing the GIS Map of Sunnyvale:

a. The end-user will be able to zoom in and out of the map to obtain the appropriate level of details.

b. The capability to add different symbols onto the map is also added, whereby each symbol represents a tree-location.

c. For each location, the end-user will have the capability to enter tree information, via pull-down screens.

d. Within the GIS Map, all the pertinent points (i.e. streets, street intersections, street names, and street trees) are linked to keys that are described by latitudes and longitudes.

e. Streets are differentiated into different categories, such as highways, expressways, regular streets and small roads.

f. City boundaries and major constructions are also included in the map.

[4901-K] TREEBASE USER INTERFACE MODULES:

a. GIS DATA ENTRY -- End-users will utilize the GIS Data Entry module to enter information regarding trees. This includes: tree type, history, location, status, etc. In case of any changes in status, end-users will always be able to update easily through this module.

b. INFORMATION INQUIRY -- End-users will be able to obtain timely "tree' information from the system by either looking up the data real-time or generating a report.

[4901-L] TREEBASE REPORT GENERATION MODULES:

a. REGULAR REPORTS -- There will be reports generated periodically regarding the "trees' information, such as the tree trimming cycle, etc.

b. AD HOC REPORTS -- End-users will be able to request reports by entering the reporting criteria directly.

c. GIS OUTPUTS -- In addition to the text reports, end-users will have the capability to generate graphical reports, i.e. the map of Sunnyvale, with tree locations printed clearly on the appropriate locations.

[4901-M] TREEBASE DATA BASE DESIGN:

a. TreeBase database must be designed with consideration of future growth and possible additions.

b. The design of the database will need to be efficient in order to provide excellent performance.

[4901-N] BENEFITS OF THE NEW TREEBASE PARADOX/GIS SYSTEM:

a. A system that can be shared/linked by all staff members within the department, via EASYLAN and modems in PHASE I of the project.

b. A tree trim cycle sub-system that can be built-in to the maintenance system.

c. A flexible, user-friendly and graphical user interface (GUI) GIS system that integrates input and output modules with the main TreeBase database application.

d. An automated historical work order system that can convert and replace the current 17,000 historical work orders on manual 3x5 inched indexed cards.

e. Portable personal computer for field data entry, data collection and graphical mapping.

f. Better information for decision-making which in turn allows for faster turnarounds and better decisions.

g. New GIS opportunities for the rest of the City operating departments, as a prototype strategy on an innovative and technological platform.

h. Improved customer services and support.

i. Tighter local management decision-making.

j. Better dialogue, ease of use, user-friendly TreeBase at the PC-level.

k. Local responsibility for users' own data accuracy and security.

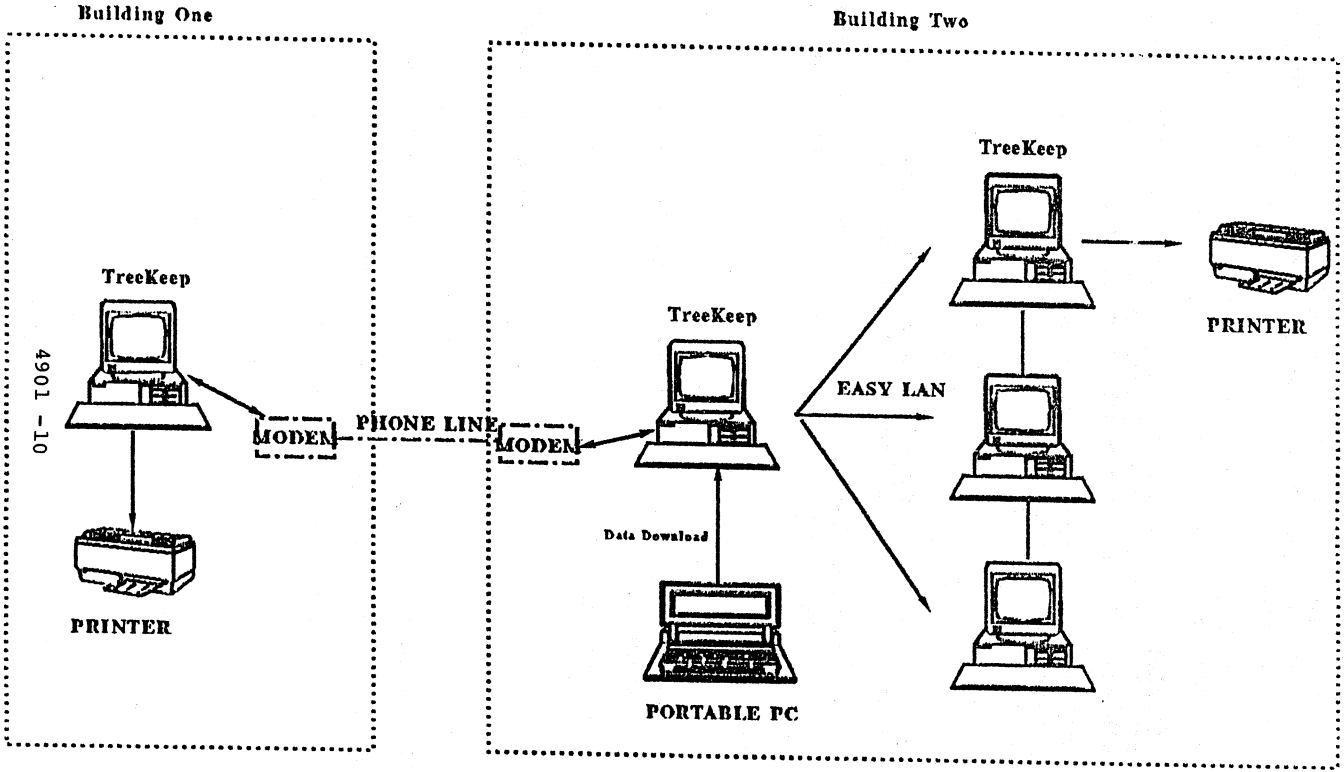
l. End-user involvement in the TreeBase system development life cycle.

m. Better control of work order scheduling process.

n. More effective use of management and work crew's time.

o. More effective employment of resources.

p. More timely management and analytical reports/graphs.


Building One
Building Two
TreeKeep
MODEM
PHONE LINE
MODEM
TreeKeep
PRINTER
TreeKeep
Data Download
PORTABLE PC
EASY LAN
TreeKeep
PRINTER

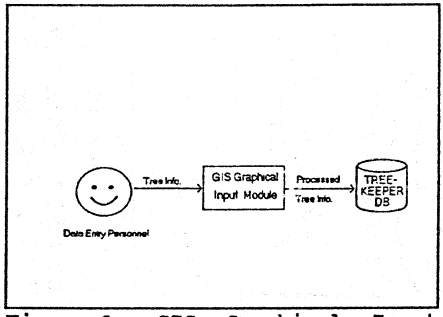


Figure 1. GIS Graphical Input Module Process Diagram

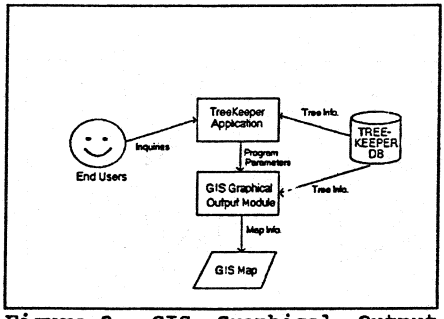
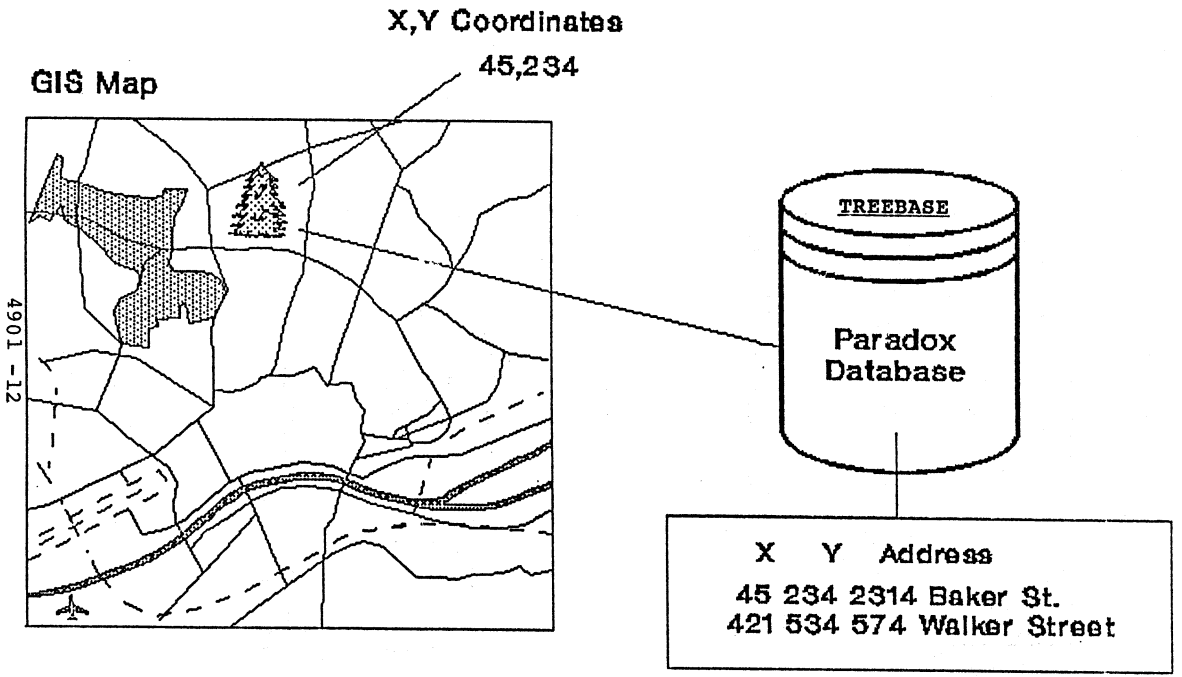
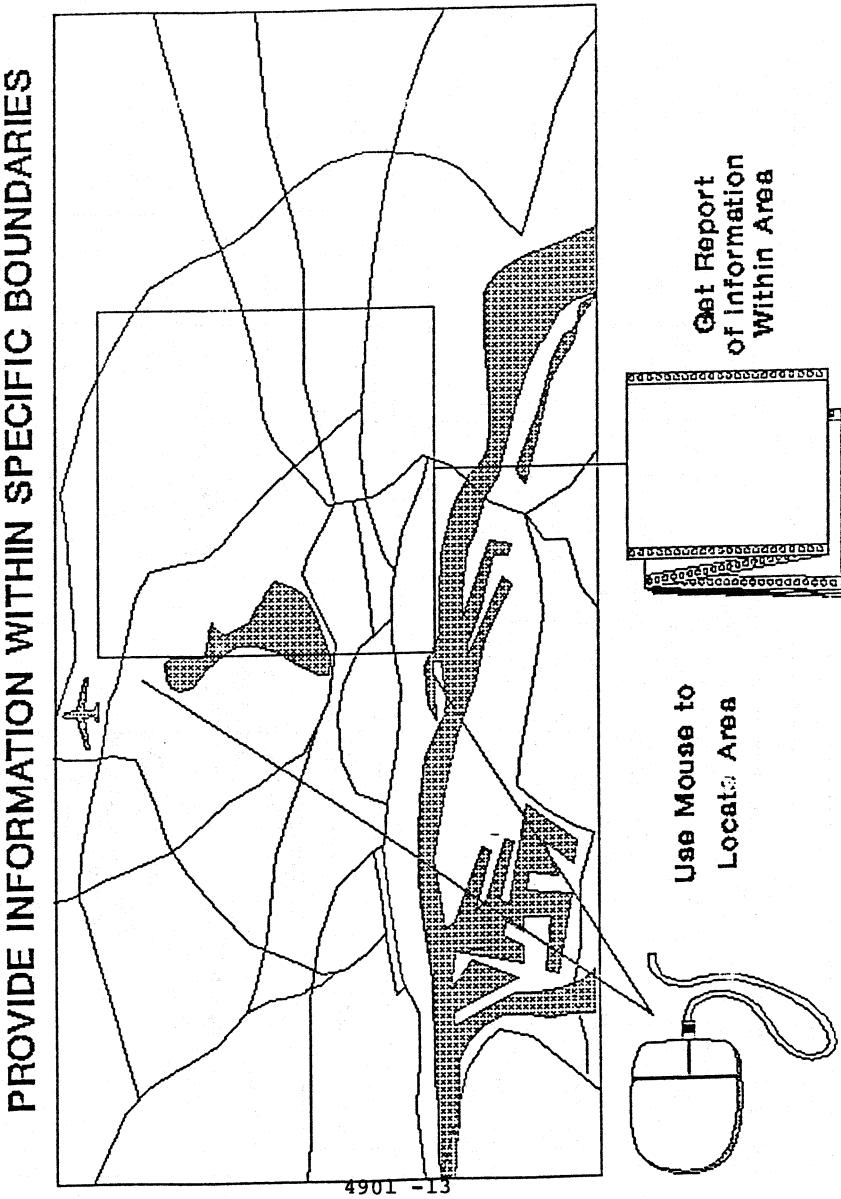


Figure 2. GIS Graphical Output Process Diagram

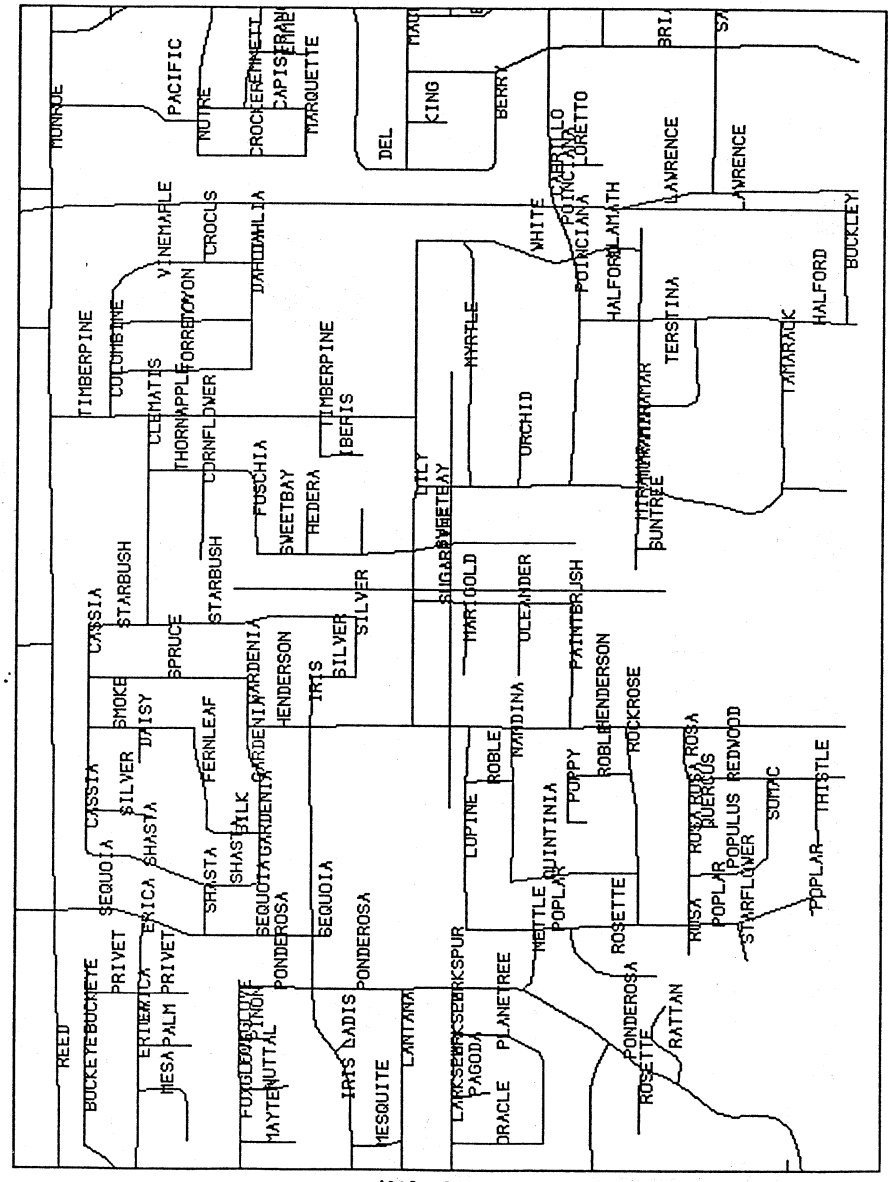


Geographical Information System

[4901-R] TREEBASE TO PROVIDE INFO WITHIN SPECIFIC BOUNDARIES



[4901-S] GIS MAP OF SUNNYVALE



IF IT AIN'T BROKE, DON'T FIX IT

J. B. Watterson
Computer Data Systems, Inc.
20010 Century Blvd.
Germantown, MD 20874
301-353-2179

If It Ain't Broke, Don't Fix It

This old proverb, once considered wise, acceptable and even prudent, needs to be updated to meet the challenges of the 90's. Consider these modifications to the proverb:

Dad's Maxim:

"If it's not broke - take it apart anyway and lose some of the pieces, then it will be."

OR

Positive Maxim:

"If it's not broke; how can you make it better?"

In the next decade, the world marketplace will be dominated by change. The challenge is for us to acknowledge that our nation, company, work environment, and even each of us need improvements for developing quality. We have to make our products and services first class again by initiating ideas, taking risks, and acting upon these ideas.

This presentation is a practical, hands-on approach to understanding and making changes. It includes illustrations which will encourage you to identify and initiate change. These initiatives can open new ideas and doors to the future for each of you.

As Louis Gerstner, Chairman and CEO, RJR Nabisco, Inc. phrased it,

> **"We need to adopt that legendary principle: No more prizes for predicting rain. Prizes only for building arks."**

Of course change is nothing new. For our first illustration, let's go back over two thousand years to 435 BC, and look at the story of Nehemiah and how he initiated change.

- Idea - Repair the wall.
- Risk - Defy the king.
- Action - Initiate action.

IDEA

Open your idea to the future and go beyond what you're doing today.

Increase Your Knowledge

Finish your degree if you haven't already. Take advantage of any courses, seminars, and brown bag sessions; read the trade magazines and newspapers; and use your technical library. Discuss technical subjects with friends and coworkers. Attend Interex conferences and your local RUG meetings. The more you know, the easier it is to initiate and deal with change when it takes place.

A man had a choice of raising vegetables in the Los Angeles area or moving to Alaska. He increased his knowledge and chose Alaska. Why? He studied the situation and decided that he could grow big cabbages and turnips in Alaska because of the length of days in the summer.

Make Decisions

The act of deciding can put you in control. Just sitting around and waiting for a change can make you anxious. You may very well decide to wait, but that's OK. At least you made a decision. Even top executives make incorrect decisions, but that's life. Don't dwell on the bad ones, concentrate on the positive decisions you have made. The objective is to take charge of yourself and do something. Listen to your gut feelings combined with logical facts.

Harry J. Sonnenborn, McDonald's first chief financial architect, stated, "We are not basically in the food business, we are in the real estate business. The only reason we sell 15 cent hamburgers is because they are the greatest producer of revenue from which our tenants (McDonald's franchisees) can pay us our rent." They still control, through leaseholds or actual ownership, virtually all the land its franchisees have restaurants on. After all these years, they are still in the real estate business, making money and selling hamburgers.

Set Flexible Goals

We don't set goals because of the fear of failure. So why not set **flexible** goals. If you realize that you aren't failing, you can be committed to yourself, making way for setting flexible goals and achieving them one by one. Consider what you *"want"* and what you *"need."* What you need is far more within your control. You won't get anywhere until a clearly defined "need" is recognized.

Make contingency plans. When generating an idea, consider if "A" doesn't work, be ready with idea "B" - create alternatives.

You are tired of going to Ocean City each summer for your vacation and you decide a ski trip in February would be better. No way says your spouse, I don't want a cold snowy vacation. Have plan B ready. Consider going to Arizona where your spouse can lounge around the pool in 80 degree weather and you can be within a one hour drive to ski in the mountains. Plan C - take separate vacations or Plan D - go back to Ocean City.

Don't over plan; everything changes in the process, anyway, so be flexible.

Express Your Ideas Freely

This is important because what's impossible today may become the norm of tomorrow. Open your ideas to the future and go beyond what you're doing today. Ask yourself "How can you make it better?"

There are no "BAD" ideas. We just need to create a vision of how the whole process could work better and how it could achieve greater results. Then share the vision and let others add to it or change it.

Edwin Land discovered the potential for an instant camera when his young daughter asked why she couldn't see a picture right after she took it. The result - the Polaroid camera was invented.

One of my staff dropped by late one afternoon and presented an idea. My first thought was "You dumb poop, that's a stupid idea." Fortunately, I didn't say that. But, when I came in the next morning, I found a sheet of paper on my desk, outlining the idea. It was a good one, I just hadn't understood it clearly. Who was the dumb poop? Yes, me.

Keep Ideas Simple and Timely

People have a tendency to back away from anything complex, regardless of how good it may be. Even great ideas won't work if the customer or your management isn't ready. Look for ideas that solve problems. They stand the best chance of being winners.

For example, Prodigy, the new computer data base, is marketed with such complex descriptions that relatively few people (approximately 700,000) understand it well enough to subscribe.

In the early 1930's, a man by the name of Carson had an idea and tried to sell his invention. No one would buy it. "XEROX!" It wasn't simple and it wasn't timely.

Think about these ideas -- who would buy a "pet rock, a hula hoop, or a yellow ribbon to tie around a tree"?

RISK

The person who does not risk has nothing.

Be Courageous

People who are willing to take a risk in presenting ideas must be very courageous. Remember in the story of Nehemiah where he looked sad in front of the king? This was prohibitive and he could have lost his head in a hurry. But he took the risk. No one is going to decapitate you. When something doesn't work, get to the root of the problem. It may be you, the system, or even someone who is sabotaging your new idea.

A Swiss chemist, George Anderman, discovered cellulose. Almost 50 years elapsed before any practical use occurred with artificial silk, under the name of rayon. Hopefully, for you, there is a much shorter period between your idea and its acceptance and implementation.

Be Positive

You must be positive, enthusiastic, committed and honest in your approach or people won't listen. Meet with the essential people in management to elicit their commitment to change and for their support. This is a function of leadership, a quality within each of you. So why not use it? The person who does not risk has nothing.

Two older men, both near death, were placed in the same hospital room. The two were divided by a permanent partition and only one man was near a window. The second man couldn't see out. The second man asked the man near the window to describe what he could see. (Close your eyes and imagine the park.) His response was, "I see a beautiful park with green grass, trees, flowers and a lake. I can see children playing, people jogging and biking, an older couple sitting on a park bench talking, and a young couple walking and holding hands. I can see the sun shining, a blue sky and the breeze blowing gently through the trees."

That night the man near the window was gasping for breath and dying. The second man thought, "I won't call the nurse as he is going to die anyhow, and when he dies I can get the bed near the window." The next morning the man near the window was dead. The other man requested that he be moved to the window bed and the nurse obliged. But, when he looked out the window, he saw only a brick wall-another wing of the hospital.

Are you going to see the brick wall, or are you going to change and see something better? Don't wait until it's broken, then fix it. Get an idea, take a chance, be proactive, do something, and make your product or service better. Be positive, enthusiastic and honest.

Work With Others

If you work with largely unresponsive people, share your ideas with the few who would be part of it. Work through them and listen to them. Put your pride and feeling of ownership aside. Be patient and thorough as people do misunderstand, just as I did in my personal example above.

Let me illustrate misunderstanding with the story of the pig.

- Misunderstanding (preconceived idea and pride).
- Resentment and anger.
- Reaction.

Make sure others understand what your idea is, work well with others and they will support you.

Take Risks and be Willing to Make Mistakes

Hope is standing up and risking in taking your commitment seriously. Sometimes though, there is a price in taking a risk. You can get shot down, ignored and even laughed at. So what? We desperately need to learn how to lighten up and live with change, or we will end up like so many already have, bearing down and getting buried.

Forrest Ford stated clearly on the subject of taking risks and expecting to make mistakes in his presentation on "Maintaining a Quality Staff" at the Interex Conference in Boston. (paper # 5081) I quote:

"Be willing to take risks and expect to make mistakes. People who do not make mistakes do not do anything and therefore never get anywhere. Progress cannot be made by staying safe. Dwell on the positive and refuse to be defeated by the negative. This takes courage and faith, but without them, many great contributions would never have been made. Dr. Seuss visited and was rejected by 23 publishers before The Cat in the Hat was published by the 24th."

There are many chewing gums, breath refreshers, sprays and toothpastes on the market today. The effectiveness of these breath refreshers can be measured in minutes. A company by the name of *Inteligis* has come up with a time-release pill called *IntiMint* that it says leaves the mouth minty fresh for about two hours. If the mint's attributes don't leave you breathless, the cost might. They are sold exclusively through mail order and cost $30 for a bottle of 200 (15 cents each). In August 1990, they sold $100,000 worth of mints. Inteligis management was willing to risk and even make a mistake by introducing, marketing and over pricing such a product.

"Some must lead and some must follow, but all have feet of clay." (Samuel Longfellow) The person who does not risk has nothing.

ACTION

Don't imitate others, be yourself - you're unique.

Forget Your Old Conditioning and Change

Don't just sit there, do it! Learn to be a Nehemiah, and not an elephant. Dr. James A. Belasco wrote a book entitled Teaching the Elephant to Dance: Empowering Change in Your Organization. We are like elephants - slow to change.

Elephant trainers shackle young elephants with heavy chains in deeply embedded stakes to make them stay in their place. Older elephants never try to leave though they have the strength to pull the stake and move beyond. Why? Because their earlier conditioning is so powerful - even when constraints to change are removed, their behavior remains the same.

Like these elephants, we are bound by earlier conditioned constraints. "We've always done it this way" is as limiting to our progress as the unattached chain around the elephant's foot. Yet, when the circus tent catches fire, the elephant forgets its old conditioning and charges. You can, too.

Recognize What Is Probable

You need to recognize what is "probable", not what you "wish" were possible. Base your actions on reality. Define what you want, and not what you don't want. Stating straightforwardly what would be best helps others to re-focus and take initiatives. Don't let a lot of confusion and misunderstanding occur. You can be successful by adapting to the rapidly changing world. Keep in mind that change for some may be tough. But elephants are learning to dance and are surviving. Won't you, too?

I took a young man to lunch at an expensive restaurant and suggested that he order something he had never had before. When he ordered a hamburger, I asked "Why?" His answer was that he had never had a hamburger that cost $9.95 before.

Phyllis Diller provided this sage advice on knowing what is probable: "Never go to bed mad. Stay up and fight."

Identify what is probable and initiate change. Stay up and fight!

Communicate, Communicate, Communicate

In the real estate business - location, location, location is primary. In effective presentation - practice, practice, practice is a powerful presentation skill. In taking action, communicate, communicate, communicate!

Very often it is not "what" you communicate, but "how" you communicate that makes the difference. "Let me tell you about a new idea I have." Positive enthusiasm for something stimulates interest and promotes in others a readiness

to accept new ideas. A negative attitude has little creative potential. To offset the powerful pull of an undesirable practice, put a new one in its place. Don't imitate anyone, be yourself - you're unique.

Do not make assumptions, particularly negative ones, about what other people feel and think. Assume that your wishes, your point of view, your actions are understood and appreciated. You will unconsciously predispose others to feel exactly that way. It's your right to have ideas, to take a risk and to present ideas to others. Remind yourself that your aim is to enhance the value of a product or service, not to destroy it. Remember, Nehemiah got the wall rebuilt in only 52 days!

In most companies, top management simply doesn't have the time to get to know most of the employees. That puts the burden on you to make the most of your opportunities. If you have 10 minutes to make a presentation to any level of management, realize that they may be the most important minutes in the total years you have worked for the company. You never get a second chance to make a first impression.

IBM's Tom Watson, Sr. once promoted a salesman he had never met to sales manager on the basis of a brief presentation.

People remember first what they heard last, but remember best what they heard first. Do your homework and prepare for the occasion as though your career depends on it --- because it does.

Add Value Constantly

Continue to add value to your services and products. Tom Peters, author of the best-selling book, In Search of Excellence, wrote, "Middle management in the 90's is a survivable profession. But only for those who take the initiative, who 'add value constantly'." Added value is more effective than a pretty face.

In your business, customers' interests should come first. How can you make your products or services benefit your customers from your ideas, risks, actions and your work? Simply, you need to be:

- o Objective - Be honest and have the courage to "tell it like it is."
- o Creative - Initiate ideas and opinions, take risks and actions.
- o Quality - Strive for quality and excellence in your services and products.
- o Value - Add value constantly to each service and product you provide.

A study by the National Science Foundation revealed that in 1987, foreigners obtained 47% of the U. S. patents, up from 34% in 1977. The Japanese led with 19% , a 25% increase over the prior year. They are not our only competitors. West Germany was up 15% and France up 19%. For the first time in history, the Japanese firms; Canon, Hitachi and Toshiba, had the most patents. General Electric, which had held the top spot for 25 years, fell to *fourth* place.

Build Customer Relationships

You need to seek an enduring relationship with your customers, based on trust and mutual respect. Your company alone can't do it, but you as employees can. A quality-oriented vocabulary is emerging that replaces the words *"user and client" with "customer."*

Again, let me illustrate with a true story on how the Japanese build customer relationships by going the distance.

- Recognize the problem.
- Added value to the service.
- Build enduring relationships with the customer.

Since World War II, Japan has been sought and gained an enduring relationship with its customers by providing quality products and services. We have come to believe that the Japanese are classic overachievers: selfless, high-tech samurai dedicated to propelling their nation toward world economic power. But in reality, the new generation Japanese may prove even more self-centered and less work-oriented than their counterparts in the U. S. Japanese now in their twenties and early thirties differ so much from preceding generations that they are called **"shinjinrui"**, or the new race.

Nearly three in five shinjinrui believe hard work and honesty do not "pay off" in their society. Among supposedly under-motivated and demoralized New Yorkers and Angelenos of similar age, the vast majority profess faith that these old virtues still work in our society. That is encouraging!

By seeking enduring customer relationships and providing quality, you will open new doors for yourself in the 90's.

CONCLUSIONS

Whether in our conscious or subconscious mind, we all have visions of the future.

Leave the Industrial Age

We are now leaving the industrial age and living in the computer age. As a result, problems are now global rather than national, and the rate of change is accelerating. The world is pessimistic - but times when the crowd suffers from the *"Chicken Little"* syndrome are the times to act. Whether in our conscious or subconscious mind, we all have visions of the future.

Challenge Yourself

Through innovative ideas, risk, sacrifice, and action, you can open new doors to the future. You can open possibilities for making changes, achieving high performance, building customer relationships and developing top quality in the products and services you produce. The result - innovated solutions to the problems of the 90's.

Go beyond what you are doing today. Don't be a cut flower person. A cut flower has no roots and it's dead. It can't reproduce or grow; just wilt and fade away. Keep your roots and stay green so you can continue to grow.

Ask The Question

Those of you who take the initiative for change and add value constantly will get ahead in the world of tomorrow. So, if it isn't broke; ask yourself --

"How can you make it better?"

References:

- Bible
- Managing Change, James McDavidson
- Maintaining a Quality Staff, Forrest Ford
- Teaching the Elephant to Dance: Empowering Change in Your Organization, Dr. James A. Belasco
- In Search of Excellence, Tom Peters

TECHNOLOGY OF DATA DISTRIBUTION

Roger Lawson
Proactive Systems Limited
Central Court
Knoll Rise
Orpington, Kent BR6 OJA
ENGLAND
Tel: 44-689-877933

ABSTRACT

One of the problems when building distributed systems based on IMAGE using HP3000 computers, is the slow response time and low throughput apparent when accessing remote data bases over NS. The author will describe various techniques for overcoming this problem so that it becomes practical to build high performance distributed applications.

WHY PEOPLE BUILD DISTRIBUTED SYSTEMS

Distributed systems can meet the needs of many organizations for more flexible and lower cost computer applications.

Distributed systems simply use multiple computers to disperse processing in a network instead of using one centralized computer. They range from the extremes of two computers placed back to back to large networks of linked computers. The benefits of distributed systems are:

- You can minimize computer hardware and data communication cost. Multiple smaller computers give you more effective power for your dollar than one large computer.

- You can minimize data communication costs. Doing local processing minimizes data transmission needs.

- Increased data redundancy can provide 24-hour uptime, good disaster protection (you don't have all your eggs in one basket) and better user response times.

- Geographically dispersed organizations can obtain local processing and local management control.

- Computing capacity growth can be more incremental - simply add another computer to expand your processing capability.

- There is more flexibility in hardware selection, computer placement, management structures, application design, etc.

However if you want to build such distributed systems on HP3000 computers using NS communication links and IMAGE data bases, you are likely to meet some performance bottlenecks unless you are experienced in building such systems. For example, doing a remote log-on to a system, opening a data base and retrieving records will all take much longer over an NS link than on a local system, as anyone who has used NS much will tell you. But you can easily avoid these performance problems by careful design and by the use of a few technical "tricks".

DATA LOCATION TRANSPARENCY

Once you spread your data bases over multiple computers, you need to be able to easily access data on remote nodes with no more difficulty than if it was on the local node. This can be summarized by the phrase "data location transparency". Users or application programs should also not need to have to worry where the data is located. See Figure 1.

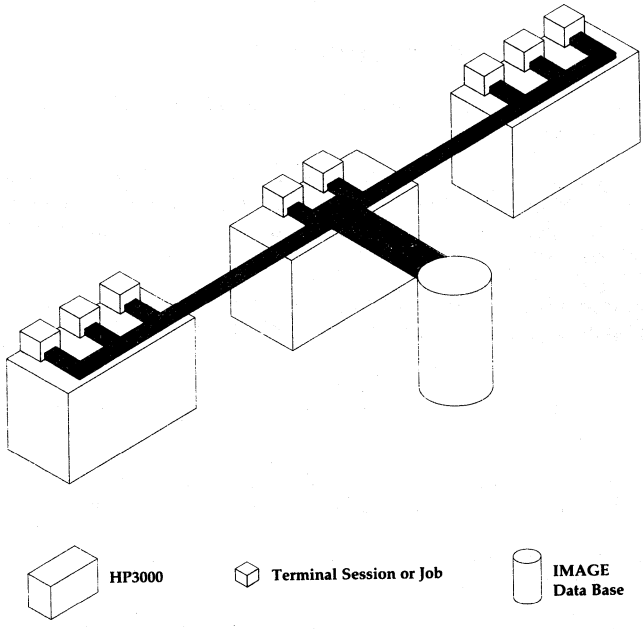


Figure 1 - Data Location Transparency Example

POOR PERFORMANCE AND COSTLY DATA COMMUNICATIONS

All of you who have used NS or DS know that the speed of access to remote data bases, both in terms of response times and data throughput, can be apparently much worse than local access. If you have a lot of user sessions accessing remote IMAGE data bases, your whole system can grind to a halt.

Remote data base access using standard NS facilities not only uses more cpu processing capacity but also generates a lot of data communications traffic - this by its very nature costs more than a local disc access. Data communication costs can be a significant proportion of overall project costs when building distributed systems. Overcoming these problems is what this paper is about.

THE SOLUTIONS

■DATA REPLICATION

Maintaining multiple copies of data can do wonders for performance. Having a local copy of a data base can be much more cost effective if the ratio of read to update access is high, as is common in many commercial systems.

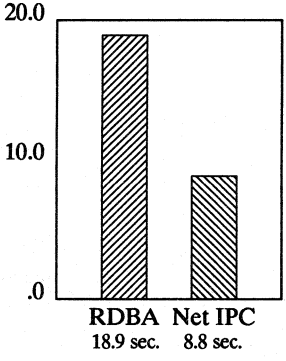
For example reference information such as price lists are best held on every node. The amount of disc space needed to do this may be small and it reduces dependance on one central copy. Any updates (which are typically low volume to such files) can be automatically replicated from the central site - possibly in pseudo batch mode if you don't need real time updating and prefer not to keep the data comm link open.

Data replication also enables you to maximize performance by using load spreading. For example you could off-load all enquiry and reporting activity to a second copy and thus improve update response on the primary. Obviously data replication uses extra disc space so there is a trade off that must be studied in each situation between the cost of additional disc space against the data communication and cpu processing cost. However disc storage costs are certainly falling more rapidly than data communication costs.

■ SOFTWARE

Another approach is to tackle the performance problem at a technical software level. Two particular performance bottlenecks that can be solved on IMAGE data bases are:

1. It takes a long time using RDBA to open a communication link and open a data base in response to ad-hoc enquiry or update activity. The solution is to hold a pipeline open to the remote data node so that response can be immediate (or at least comparable to local data base open time). Figure 2 shows some comparative figures I obtained comparing the two methods on our development systems.



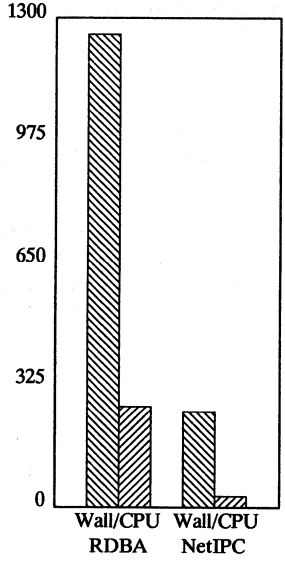
NetIPC versus RDBA

Elapsed time in seconds to establish remote session, open data base, read three records from master set, read one detail set record and close data base. Average of twelve runs accessing MPE/V system from MPE/XL system.

Figure 2

The pipeline can be shared by any number of users and can simply be created when the first user connects.

2. Poor throughput or high response times can be caused due to the large amount of data transmitted by RDBA over communication links which are always relatively slow in comparison to a local access path. For example RDBA transmits the whole data base buffer which is created on a remote node. However typically you don't need all the data items in all the records. You probably only want some of the data items from some of the records. Solution: do the processing on the remote node (where it is local to the data base) and only transmit the data required by the user program. Figure 3 shows some comparative figures I obtained.



NetIPC versus RDBA

Time in seconds to perform a serial read of 7000 entries from master dataset residing on an MPE/V system from an MPE/XL system via NS.

Figure 3

Both of the above techniques can be implemented in a generalized manner and we did this using the NetIPC service of NS, ie. you don't have to change the application software to get these kind of performance improvements. By using NetIPC and appropriate buffering strategies you can effectively optimize what you get transmitted over a network which is the key to performance maximization. Making the best use of each NS transaction is the key!

■ SYSTEM DESIGN

You should process the data where most of it is located so as to minimize the amount of data transmission over the network. Consider the following example: you have two computers named A and B with the users logged onto A but most of the data they access on B - they want to run a report which simply wades through the data base to produce a total. It is much more efficient to run the reporting program on computer B and route a few report lines back to A, than run the program on computer A in which case every data base I/O goes through the computer link.

You could use remote process handling with the pipeline architecture mentioned above to achieve this kind of result transparently to the user.

Using the above data replication and software optimization techniques can reduce the data transmitted and hence the cost by a factor of up to 10. Also careful consideration to optimal data communication configuration (which is outside the scope of this paper) also pays good dividends.

As always, careful application design is also a key factor in minimizing data traffic and maximizing performance.

With distributed systems you need to spend more time on careful design of the application system because every I/O in a network "costs" a lot more than the same local I/O.

CONCLUSION

To summarize, building high performance distributed systems is possible on HP3000s. Using the techniques described above you can construct applications such that overall costs are lower than they would be in a centralized system, without having to accept lower response times.

Paper #5104

The Effects of 4GL (Powerhouse) on the MIS Environment

Brad T. Smith
Rochester & Pittsburgh Coal Company
655 Church Street
Indiana, PA 15701
(412) 349-5800

I. Effects of PH on Programming

Comparing PH programming to Cobol programming is much like comparing walking to driving. The advantages are quite dramatic. Programming systems entirely with PH gives you the potential to organize your programming effort differently. I think it is important to understand what changes you can expect, how they can benefit you and what implications these changes have to a manager.

1. The first is, of course, the sheer reduction in programming time. One day's effort of Powerhouse programming will typically produce what one month's effort would have in Cobol. Just the reduction in coding line volume alone is considerable. Expect about 15 to 1.

2. You're going to need a lot fewer programmers for any given work load.

3. Programs will cost much less to produce.

4. You're going to be able to reduce your backlog much, much faster.

5. PH programming is much easier. The training effort and learning curve for new programmers is much sharper and technical skills are much less critical. Lack of "senior" programmers should not be a bottleneck and new programmers can be doing productive work shortly after their initial training. In fact, because of the greatly compressed programming cycle, analytical skills will be more important and more noticeable than ever before.

6. PH programming is far less intense, requiring less concentration for shorter periods of time to produce higher quality programs. (You can relegate burnout and stress to stories you tell your new programmers to amuse them.)

7. Dictionary control eliminates vagueness and oversights. Data consistency is guaranteed throughout your programs.

8. PH programming is virtually bug free, requiring less time and skill to test. This aspect alone is worth a great deal in peace of mind. New systems are no longer the "risky business" they once were due to bugs. And let's face it, testing is one of the most neglected areas in programming. No one likes it and very few programmers, experienced or not, do enough of it. I'm not suggesting that bugs can't happen. Features still need to be tested to see that they were included, but not structures. And your PH programs will virtually always cycle at once. Believe me when I say life will be much less exciting than when bugs could slither out of even mature coding; weeks, months or even years later.

9. Truly structured programming. Something everyone talked about but never really succeeded in enforcing (or even understood). This

means that all those promised benefits of "structure" are going to be a reality. Programs can be easily enhanced or modified without days of study, and programmers can understand other programmers' work (and, indeed, their own afterward).

10. Because programs are much smaller, and for the most part deal with results rather than procedure, they are adequate as technical documentation. No other documents are really needed to preserve a complete understanding of the program.

11. Invoking on-line user documentation is very straight forward and has been built into the structure of PH. While it may not, in all cases, provide your entire user documentation, it can in many small or less involved systems do just that, and go a long way toward making larger systems self-documenting.

12. Some other things you can expect to eliminate to your advantage are: flowcharts, compilers, object code, special editors, debuggers and the other special tools that seem to abound in procedural environments. Gone too is the confusion they inevitably cause.

II. Some Thoughts on Organization in a PH Environment

Because PH programming is much faster and far less abstract, a programmer can be expected not only to do more programming, but do much more of the entire process of implementing systems. Consider the following job description, which incidentally is from our organization.

JOB DESCRIPTION

TITLE: M.I.S. ANALYST/PROGRAMMER

Generally responsible for the programming, installation, maintenance and support of computerized systems.

Major responsibilities include:

1) Aid users in the analysis of manual systems to be computerized.

2) Analysis of offline and supporting activities effecting the ultimate use of computerized system.

3) Aid in functional system design.

4) Aid in preparing technical design specifications.

5) Programming

6) Program testing and debugging

7) User training

8) Technical documentation

9) User documentation

10) Ongoing support and review of systems.

and other related activities as may be required.

Notice that I've described an Analyst/Programmer. Pure programmers are non-existent in our shop. With this vertical arrangement, I feel we are able to install better systems significantly faster.

I realize that this job description implies not only technical skills, but even more importantly, people skills and analytical skills. It is not enough that analyst/programmers merely understand how to program. They must really understand the corporation, what it does, how it does it, why it does it and, more importantly, how it wants to do it. With PH, people who possess analytical skills can more easily master programming skills, because they are so much less demanding and time-consuming.

Another very important consideration when using a vertical approach is to use only one analyst/programmer on a system. Believe me, one head is much better than two. With fewer people milling around, timing and coordination is not a problem. Sense of responsibility, sense of accomplishment and work pace all improve dramatically. Even very large projects can be handled this way. In fact, the larger and more involved project, the more you'll appreciate the benefits of one head.

One other point to consider, No. 3 and No. 4, aid in functional system design and aid in technical design. By that I don't mean to confuse the design of a system with the implementation of a system. I feel that the analyst/programmer should be included at all stages of the functional design discussion (especially with the users with whom they will work during installation). And certainly they should be involved during the technical design of the data base, menus and program specifics (here their technical experience and expertise will be invaluable to the designer). But unless you have experienced designers working as analyst/programmers, I wouldn't combine those functions. (All too often programmers, by default, are put into a position of having to design the system. This rarely works, and at best, usually produces mediocre and untimely results. Too many organizations blame programming for failure or delay (and indeed routinely confuse programming with design) when the fault is really the lack of reasonable design effort.

This vertical combining of related skills represented a major departure from the way MIS is traditionally organized. It has worked exceedingly well for us, and I highly recommend it.

III. PH As a System Designer's Tool

While PH is generally perceived as a programmer's tool, it can also be a very significant tool for the system designer. The system design is still the single most critical element of a successful system. Bad programming can be repaired eventually. Bad systems can't. Good programming can't save bad systems. Make no mistake about it, PH is not going to make you a system designer. But it can be an important new tool for you.

Perhaps the 5th generation of software will turn ill-defined problems and desires into "structured", programmable solutions. But for now, it is still up to a designer to insure that the programs he specifies are an appropriate solution to the problem at hand.

Faster, more accurate programming, by itself, doesn't necessarily speed the design process that proceeds it and specifies it. Nor does it necessarily cause the design to be better. But because PH programs can be produced much faster and more accurately, and modified more easily than traditional programming, they allow the system designer a wide latitude of use during the design process.

Faster, better and more flexible designs can result with newer strategies using PH.

Some Persistent Myths

Traditional design theories, while differing somewhat in detail, have as their central theme the idea that if only we would spend more time up front, we could design better systems. Better study methods, better questions, better meetings, better working specs, etc., etc. All this activity contributes mightily to the lead time of an application. In many projects it can easily amount to 50% of the total time spent on the entire project and an even higher percent of the "elapsed" time. Worse yet, the penalty for not designing carefully up front was to waste that traditionally scarce, laborious, difficult undertaking, programming.

Let's be candid about this process. In spite of our best efforts and even better intentions, designing has been far more promising in theory than it has ever been in reality. In spite of all attempts, rarely can we protect programmers from wasted effort.

Face it, while it is probably possible to design an acceptable system up front, your chances of doing so are practically nil.

Let's Look At The Realities of Design

The four awful truths of Systems Design are:

A) Very few potential "users" remember all aspects of what they truly do.

B) Fewer yet know why they do it.

C) Even fewer than that know what other departments around them do ...

D) Almost none know a better way.

Users confronted for the first time with specifying their needs and desires for a new system can usually be heard gasping for air and making inaudible noises. And if the success of your design relies upon these people defining something you can hang your hat on, and it usually does, you're in trouble.

Experienced designers who have participated in this ritual quickly realize that they'll be lucky if users know what they need, even after seeing it! Simply put, the lack of "understanding" is the major time and quality constraint in designing most systems.

There is an Easier Way to Design Systems

Like it or not, successful systems aren't so much designed into existence as they are criticized into existence. As you have probably already noticed, users are far, far better critics than they are specifiers, and criticism is a much faster and more natural way to create systems. When there is something concrete to object to or point to or grieve about, things tend to move faster. Without something real to look at and touch and be critical of, the average user is pretty disoriented and confused by what he perceives as an abstract technical problem. He needs a focus to work from.

Even designers work empirically. Every design starts somewhere early on with trial assumptions and is discarded or refined and finally

implemented. PH, if you let it, can turn what used to be nasty liabilities into assets.

Because PH programming is so much faster, it can be invoked much earlier and become part of the design process itself. Now I am not suggesting that you simply blaze away with programs on a trial and error basis. That too can be very frustrating and wasteful. Nor do I quite agree with so-called prototyping. Prototyping, to me, is different in a critical way. It's not the real thing and users know it. They don't give it the attention you're looking for at this point, and as a consequence, it does not "ring out" the design as intended.

(I consider prototyping to be a halfway measure. A throwback to the days when the real programming was a very lengthy and costly endeavor. I must confess I've designed lots of systems that got tried and thrown out or drastically altered, and in effect, were prototypes, but I don't believe in the deliberate creation of these things.)

I tend to look at PH programs as "living" system specs and only slightly more involved than a good system spec. So why not do the real thing? Instead of slaving over and presenting a spec (which few users will read and understand anyway) why not present, instead, a real system and get on with the business of using it or modifying it.

I introduce PH programs at about the same point I would feel comfortable with a first spec. We program only enough of the system to prove our point. That is that:

A. The design is going to match the user's expectations and abilities, and ...

B. The data base is really going to cover the needs of the system.

Usually the dictionary is very basic at this point, and usually, we're only into critical screens. Security, reports and batch work can wait, as they are more easily envisioned. The trick is to invest just enough programming to verify your system's workability and acceptability without investing more than would make you cry should you have to ditch it.

Design Flexibility

One of the more important attributes of a good system's design is its ability to be modified without coming apart. If the idea of introducing "real" programming earlier on in the design cycle is to work, it must be easily modified. Indeed, it will almost certainly need to be modified.

Because PH programs are inherently structured, to modify them is a very straight forward process. And unless you data base is badly off, you can very quickly respond to change and try again. Adding fields, or even files, is not very traumatic (we use "Adager" very extensively), nor is adding features or even programs.

In short, while you may never entirely learn to enjoy criticism and modifications, it really can become almost a non-event with PH.

Faster Technical Design

The very "building block" nature of PH is going to help you assemble a technical design for your systems faster. Each PH application implies, by itself, a basic design spec, i.e. Dictionary, Quick, Quiz and QTP each implies a basic "building block" of activity. Also, within each of these blocks is a very "structured" approach to the particular type of logic they support.

The Effects of 4GL (Powerhouse) on the MIS Environment

These PH attributes mean that as a designer you can assemble, very quickly, the basic "blocks" of logic you need to solve your design requirements, and within those "blocks", specify the behavior you desire in almost "checklist" fashion. As a consequence, you will spend much less time envisioning specific program procedures. You know instinctively that they can be covered easily within the confines of the block you have chosen. After a few PH systems, the process becomes almost instinctive and very fast.

Modifying designs is equally easy. The process of adding blocks of activity or modifying the functions within them becomes second nature.

My experience with PH is that the technical design, with practice, came much easier to mind, was much "cleaner", more easily modified and "aged" well. Structure in design is as important as it is in programming. It really does help if you let it.

Faster Real Results

Another time saving benefit you can expect as a designer is the ability to communicate easily and rapidly to your programmers about what you want, once you have decided programming is appropriate. Usually, we prepare a simple data base sketch showing files, keys, relationships, type and major content and a menu of critical or mainline programs and which applications (Quick, Quiz or QTP) they are to be programmed in. Depending upon the degree of difficulty I consider each program to be, I provide function notes on them and discuss them verbally with the programmer.

At that point, programming can begin. I know that unless something very difficult or unusual in design has gone on, or the programmer is very inexperienced, he can't stray too far from the desired result.

Any subsequent modifications or enhancements (and there will be!) are done much the same way. Layer by layer the design evolves until, finally, the entire system is approved and goes on-line.

The speed with which all this activity can be accomplished is important to quality as well. The less total elapsed time this process takes, the fresher everyone's thinking is. In our shop days, not months, have elapsed, and very few details or desires are forgotten as a result. Users actually remember what they asked you to do and what the system is all about (amazing).

And never underestimate the power of (nearly) instant gratification. PH is truly "management friendly"!

IV. Powerhouse Considerations

Designing for PH, like designing for any other language, definitely requires a knowledge of the tool. To presume that designers need know very little about PH, or that program specs will always translate from another language to PH, would be a mistake. By all means understand PH before attempting to design for it! Even if you're not going to do the actual programming, don't saddle PH with a bunch of misconceptions about what it will or won't do. You must also understand your op system's data base and file constructs, because that's what PH uses. If you don't understand both these things, you're not ready to design for PH.

Every tool has its strengths and weaknesses. I think all experienced designers instinctively meld the strengths into their designs and avoid the weaknesses (or perhaps forget them). PH is no different. It has things it does extremely well, things it does not so well and things that are downright ill-advised or impossible. As a matter of course, we never

dwelled very long on what PH wouldn't do. We were too concerned with what it would do, and we found that to be quite considerable and quite adequate.

The Design Process

Nothing has changed. PH is not going to design your system. You must still do that. So be careful. PH programming is so easy to use, it's always a temptation to start programming too soon. Resist that temptation. Do the proper design workups before anyone touches a computer. This is probably the most common mistake in our business. I've made it, I certainly see it made a lot, and I'll bet you've made it too. Design slow, program fast is an old axiom and I'm still a firm believer in it.

You'll progress faster and easier in your use of PH if you keep your first PH systems simple. Enhancing existing systems with additional reports or screens is an excellent way to start.

After working up the functional aspects of your system, the technical design process itself is not going to be much different from that of any other language. That is, in some fashion, you're going to:

a) Specify a data base.

b) Lay out a "menu" of programs to use and maintain the base. Assign each program task to Quick, Quiz or QTP.

c) Make appropriate notes on what each program is to accomplish.

and then, before any programming starts,

d) Discuss the design with whomever is going to program it, and agree that it is indeed doable (all the while, of course, keeping PH abilities firmly in mine).

There are many design considerations in a Powerhouse environment.

1) Keep a strong user involvement throughout the design stage. This is especially true when working with first-time user who may be defensive, confused, or even frightened.

2) Avoid unstructured problems. In these cases users have no solutions, or may have an elegant solution looking for a problem. Committees are usually a tip-off that this type of situation may be brewing. These typically turn into BS sessions in which very little real progress is ever made.

3) Demand offline support from the User Departments. Strong systems and business understanding must be or be made adequate. In financial applications, both manual and automated controls should be examined, installed, and enforced.

4) MIS must take time to understand the total environment of the problem and the proposed solution. What other departments are involved? Are there possible interactions with other systems?

5) Understand that you may have to develop the solution. In many cases, the users haven't a clue as to what is needed, or even possible. Don't ignore the low-tech or no-tech solution. In some cases these may be more appropriate answers.

6) Simplify, simplify, simplify. In most cases our first impulses are usually complex. If we are confused at all about our systems design, imagine how the user will react.

7) Use a "building block" or phased approach to systems installation. Do the easiest, fastest part of the system first. Especially where new users are involved. This will build confidence in the system's future, as well as the user's confidence in themselves and MIS.

8) Systems must be made fail-safe. Back-up and recovery considerations must be well thought out ahead of time, before your system fails in the middle of the huge mass update to your accounts receivable files.

9) Avoid programming while designing, and avoid designing while programming. Our shop definitely uses the "design slow, program fast" philosophy.

10) Where possible, use simple programs. We shy away from "gee whiz" or "macho" programs that purport to do "all" in one fell swoop. Keep programs easy to understand and take advantage of the Powerhouse language wherever possible.

11) Your systems must be rugged. They must be able to withstand even the toughest, meanest, smartest users you have.

V. Questions I am Most Frequently Asked About PH

During the ten years now that we have used PH, I have had the opportunity to talk to lots of potential users. Here are some of their most frequently asked questions, and perhaps the answers will help you to better understand PH's potential impact on MIS.

Can anyone learn to program Powerhouse?

Virtually any reasonably intelligent person who wants to and will take the time can probably learn to program in PH. Really simple programs in Quick and Quiz can be mastered in minutes. I hasten to add, however, that in my opinion, PH is a professional's tool and not designed for casual use by amateur programmers. To use PH effectively requires training, experience and support. Make no mistake about it, PH is a completely new and very extensive set of skills that are demanding learn. It's not going to jump into your lap and kiss you.

Have you let users program?

Yes, but very cautiously. The need for user programming to relieve our programming staff has not been a concern of ours because PH is so productive. (Normally our backlogs are very slight, almost week to week.) In some areas where we wanted an hoc user reporting capability, we tried Quiz briefly and found that unless the users were extensively trained and used it routinely, they soon forgot a lot about it. And while simplistic things are easy to program, what they wanted was usually more involved.

As for Quick and QTP, I would never consider giving end users these kind of tools. Their use gives the ability to create and alter data outside the realm of controlled systems. So unless you have user personnel who's job specifically includes system design and programming and they have had proper training, I would not recommend giving them PH.

Ad hoc user reporting for one time requests and what if stuff is a different matter, and we use it in selected systems. But it is not pure

Quiz. It is a hybrid system that we developed using Quick screens to ask "report" questions and a Quiz program to generate a "user" Quiz program. We have used this approach carefully for five years now and have had good success with it.

What kinds of systems can reasonably be done with PH?

Virtually any "commercial" system that you would normally tackle with Cobol you can do with PH. There are a variety of systems done via PH at our Company. They range from financial applications to Geological, Production Reporting, Safety, Engineering, and so on.

Obviously very special tasks, like word processing, plotting and drafting, spread sheets and such, we of course use more specialized tools. Also very heavy "number crunching engineering stuff" our engineers do in Fortran. (Interestingly enough, we found that a large percent of what our engineers do is nothing more than good old-fashioned data base maintenance and data processing, and of course, we do all that in PH.)

Realize too that our system philosophy dictates that all operations are to be performed by users. Operationally, MIS does backups, the occasional tape mount and virtually nothing else. So "user friendly" is more than just a buzz word to us. We live on our systems' ability to stand the careless abuse that only real users can give them. We found PH to be exceptionally well suited to this "hostile" environment.

Will systems have to be designed differently to use PH?

Perhaps somewhat. The functional expectations for your systems should not have to change. In fact you may find, as I did, that not only do you not have to compromise your expectations, but you could get some better ideas on how things might be done. The technical design, that is the data functions of programs you might specify, could change somewhat.

Will PH programs require more processing time?

Probably. I almost didn't include this question even though it is very commonly asked. To be perfectly honest, I don't know. I really don't have the inclination to extensively "bench test" performance, and I really don't have figures or facts to give you. Having said that, what I can offer, for what it's worth, are my observations of each of the program elements of PH.

a) Quick Screens

My comments in this case apply to character mode screens, as we do not use block mode Quick screens. Quick, again in my opinion, executes as fast as any well written screen program and the processor load (with performance evaluation tools like OPT) appears to be no different than Cobol screens. Character mode Quick screens appear to execute faster and require less processor time than block mode "View" screens. Unless you have an extremely unusual situation, I wouldn't worry about them. In short, QUICK IS QUICK! Our users demand and get very snappy response time.

b) QTP and Quiz

QTP and Quiz are probably less efficient than tight coding in Cobol or RPG. Remember that what they are doing is probably going to be sorting, and sorting is sorting. I consider the times for our average runs of say several thousand records to be inconsequential.

Having rather begrudgingly said all this, I think it is appropriate to address the tendency to optimize computer processor

performance at the expense of far more costly resources. That is to use more programmers and more clerical personnel (in other areas) while incurring longer lead times in systems development and backlogs of undone systems. And of course, all the while, losing the business opportunities of better, less costly systems and information.

All this to presumably save a far less costly resource, computer processors. Currently, processors typically cost less than the libraries they work with and much less than the people that program them. In my considered opinion, the very reason businesses buy computers (to use them!) shouldn't wait while some misguided attempt to get the Nth degree of performance out of them grinds on.

This performance focus appears to be a throwback to when processors were much less powerful, much more expensive and applications were fewer in number and were executed one at a time.

(It has also been my experience that a lot more performance has been lost in sloppy system design than has ever been gained in efficient programming. I've seen systems so badly structured that they were "dogs" and no amount of clever programming language was ever going to make them otherwise.)

Even if there was a significant PH performance difference, which in my experience there is not, the overall benefits of PH would still far outweigh this consequence in most MIS environments.

Will experienced programmers find PH programming frustrating?

I'm not sure, but they shouldn't. I think at first everything that is different tends to be perceived as more difficult or restricting. (And make no mistake about it, PH is different. If it weren't it wouldn't be worth bothering with.) I've never really had "experienced" programmers working with me as PH programmers, but one five year RPG veteran felt there had to be a better way. He found the transition refreshing.

I suppose to some programmers there could be a sinking feeling that perhaps they're not needed, or it's no longer "real" programming or many of their hard won skills just got obsolete. But, in the long run, the case of PH will outweigh these concerns.

VI. IN CONCLUSION

The Shoe Maker's Kids Get New Shoes (Finally)

Never have corporations had a greater need and higher expectations for automation. Never has so much emphasis been placed on making computers truly integrated parts of business. Survival in the 90's, for most businesses, will include the routine use of computers, everywhere they can and in the best ways they can. Faster, bigger (and smaller), cheaper hardware abounds and will continue to proliferate. Terminals are practically on every desk, real time has replaced batch, data bases have replaced mere files, data networks are routine stuff and on and on.

Yet a significant number of computer groups on the so-called "leading edge" of all this technology and who are expected to be fearlessly leading the charge to greater efficiency, greater productivity, greater and greater cost savings (not to mention world class excellence) are still using 1960 tools.

I find that truly amazing.

What other high tech endeavor could possibly be expected to survive the 90's using 25 year old tools and even older methods?

But there is a better way. And I hope that's what this paper has been about. Like most worthwhile things, the 4th generation is not free. It will require effort and new skills. But I can assure you, from my own personal experience, it will be very well worth it.

5105

MAINTAINING A QUALITY STAFF STARTS DURING THE INTERVIEW

JEFF ODOM

BAHLSEN INC.
ONE QUALITY LANE
CARY, NC 27513
(919)677-3227

"One job belongs to the leader alone; and that is making sure all the parts and all the people work together."[1]

You can have the best parts in the world - yet the truck you build may not run well, if at all. You may have the best flour, sugar, spices, oils, etc., but the cookies you bake may not taste good and may be hard as a brick. How is it people can shop at the same supermarket, buy the same ingredients and use the same recipe yet the final product be as different as the east is from the west? The functioning of people, their interrelationships, skills, motivation and quality consciousness bring the ingredients together and produce quality goods and services.

We are responsible for blending associates to create a quality team. "In a system everything has to work together harmoniously. The parts and the people have to be optimized to work with each other toward a common goal."[2] You must be sure to fit the right person into the right position by viewing the whole employment setting and pay particular attention to intangibles. Success will always come down to intangibles of fit, personality and relationships.

Unfortunately, key organizational ingredients may be missing if your company does not subscribe to philosophies that create fertile environments for associates to develop. Even so, you must do what you can to create a divisional, departmental or workgroup environment which will promote such associate growth. Some of these concepts include full employment without fear, training, quality, mobility organizationally and so forth. People want to perform in a manner that provides them the satisfaction of a job well done. The work place must allow associates to perform to the best of their abilities without fear of reprisal and be fully trained to perform duties as assigned.

Quality work relationships, products and services depend upon individuals filling appropriate roles on a team. Our world is full of examples; such as consistent top flight college

basketball programs built on quality team defense. Another favorite example is of an orchestra. In an orchestra, each member is a qualified and talented musician, but to perform as a team each must fill their role at the appropriate place and time under the conductor's leadership (coordination). If each member of the orchestra were to try and highlight their individual skills, the harmony which creates the beauty of the music is lost.

"Quality is pride of workmanship or joy of work. By allowing and even urging workers to experience intrinsic rewards that come from doing something well, using their innate and acquired abilities, productivity improves, quality improves and customer satisfaction improves."[3]

Our responsibility is to place the right individual in the right position. Further, create an environment that allows for pride of workmanship and personal development. By doing this, maintaining quality staff will happen naturally. This is not setting high standards and sticking to them often demanding the impossible. Nor do we have to reward excellence and punish so called mediocrity. We must empower the individual with dignity, knowledge and skills so that they may contribute to our team.

So, what is your environment? Corporate? Divisional? Departmental or Work Group? Obviously we hope, all the way up, it is an environment of individual empowerment, quality, cooperation and teamwork. It just makes sense that this is the environment of success. Varying combinations at different levels of your organization have to be factored in. Our hope is your organization will come to terms with this because truly this is what we all want as candidates for positions.

Understand that this organizational environment or system is key to associate success. The most important factor in any hiring decision and success of a candidate, no matter what the degree of the appropriate organizational environment, is the personality fit of candidate/associate and immediate supervisor. Talent, technical skills, motivation, education and so forth are important qualities but the ultimate success lies in the relationship between supervisor and subordinate.

We may say that the keys to successful hiring are:

1. Associate and supervisor fit
2. Organizational philosophies
3. Skill, experience, education fit
4. Team fit

If you notice, we as management, control or highly influence points 2, 3 and 4. We, to a varying degree, influence corporate philosophies and should be responsible for our own

team development. As for point three, we select the appropriate skills or they are obtainable by providing education and training to associates. All this leaves us with two dilemmas of making sure we truly have the appropriate skill set and more importantly how to determine supervisor/associate fit.

May I suggest a methodology that has worked very successfully for me and was derived in large part by the teachings of Dr. Kurt Einstein noted behaviorist. Dr. Einstein taught his methodology in a seminar all over the world called "Picking Winners." The following is how I use his method incorporating processes influenced by other behaviorists as may be noted.

The basic process is:

1. Define the position to be filled

2. Determine a candidate's
   a. must haves
   b. preferred to haves

3. Advertise or network for candidates

4. Preinterview planning

5. Interview

6. Second interview if necessary - or is it always necessary?

Defining the position then developing the must and preferred to haves is critical to the selection process. The definition determines who will be interviewed. We don't want to exclude a qualified candidate and we also cannot spend precious time interviewing candidates that do not fit our needs. Further, once hired, this definition is given to the associate as a clear representation of what they will be held accountable for.

To define the position we must determine the following:

1. What will the person do?

2. What will their responsibilities be?

3. What decisions will the individual be allowed to make in 3 months, 6 months and a year. This is an important process in developing associate/supervisor trust. Robert Tannehill wrote of the importance of this point as it pertains to managers:

"For an organization to give managers freedom to make mistakes, to grow and to learn, there must be a fairly high level of trust and confidence among the managers one with another, and especially on the part of top management. When a mistake is made, it is often the practice of management to scurry about to try to find out who is to 'blame' for the error."[4]

I always tell associates immediately you have the right to make mistakes, your obligation is to communicate to the appropriate individual(s) as soon as you realize the mistake. By setting out clearly when decisions are the associate's responsibility, there is comfort and greater ease in the decision process.

4. What technical skills and education does the position need?

Having done the necessary to understand our position, received and reviewed resumes and set appointments with qualified candidates, we move to the critical process of interviewing the candidates. The interview process is an evaluation process - there is a mutual decision being reached. Both the candidate and the interviewer (organization) have to reach a quality decision as to fit. Dr. Einstein taught in his seminars that there are two reasons why personnel selection decisions fail:

1. Failure to evaluate fit of selected person.

2. Failure of selected person to evaluate fit (confusing specifications or poor understanding of specifications).

If we understand what causes personnel decisions to fail, we then can define the goal of the selection process. Simply, our goal is to avoid the failure issues by increasing our ability to make predictive selection decisions. To do this we want to evaluate:

1. The capability of the person. What a person can do leads to performance. What a person can explain to us in detail they can do.

2. The personality of the person. What a person is like leads to behavior.

Knowing these two points, we will be able to predict how a person will perform and act.

Prior to the interview, review the candidate's qualifications to determine strengths, weaknesses and fit to your criteria. Determine appropriate questions that will validate the strengths or true attributes and the extent of perceived weaknesses. Remember we all have strengths and weaknesses so our objective is to make an informed decision as to the candidates fit given both. We are not looking for someone who has no weaknesses.

To make this evaluation, we want to conduct the interview as follows:

1. Set a comfortable environment overtly free from interruptions and distractions.

2. Ask for agreement in the process. Ask may I take notes? Do you mind if we sit here?

3. Explain that this is a mutual decision process and the penalty for error is high for both parties. Further, both parties are responsible and accountable for the decision.

4. Direct the interview in a fashion that you ask all of your questions first then allow the candidate to ask any and all questions. Have questions flowing one way.

5. Ask open-ended questions. Avoid yes and no responses.

6. Ask why, why, why! Probe pat and standard response by exposing the reasoning behind them.

7. Ask revealing questions as described below.

One of the problems in interviewing is determining what questions will stimulate responses that will most add value to your decision process. What is proposed here is that you ask open-ended questions to which the response will tell you about the true capabilities and personality of the candidate. Below is a list of example questions from Dr. Einstein's seminar as well as those other successful interviewers in our organization have used.

1. What would you describe as the best attributes of your current supervisor?

2. Tell me about two serious interpersonal relationship problems you've had on your job(s).

3. What are your likes and dislikes? These are from all phases of life; personal, emotional and career.

4. Give some examples of situations where you have been criticized. How did you react and why?

5. Under what conditions do you feel you learn the best?

6. To what extent do you feel that your college grades accurately reflect your ability?

7. What do you feel are your three most significant accomplishments? In detail describe how one of them was achieved.

8. Describe the perfect boss.

9. Under what conditions would you consider yourself as "having arrived" in your personal and professional life?

10. How do you resolve family differences?

These questions give you a flavor for the types of things to ask to discover who the person is and what their capabilities are. For technical aspects, you would key in on accomplishments and the detailed description of what took place. For supervisors, look for questions that reveal personal behavior skills (so called soft behavior). In all questions, continue to seek answers which will allow you to make the best predictive selection decision you can.

The complexity of interpersonal relationships is vast. Our probability of success as organizations increases as we select the best candidate for a given position. It is important that we continue the development of all associates once hired. Give your staff an opportunity to take responsibility of their own future by empowering them with training and security.

---

1. Rafael Aquayo, Dr. Deming the American Who Taught the Japanese About Quality (Carol Publishing Group, 1990), p. 180.

2. Ibid.

3. Ibid, p. 45.

4. Robert E. Tannehill, Motivation and Management Development (Auerbach Publishers, 1970), p. 81.

Effectively Managing Your System's Resources

Paper 5106

by Carl M. Rusk
Infocentre Corporation
600 E. Las Colinas Blvd. Suite 1920
Irving, TX 75039
(214) 556-2355

Introduction

When it comes to managing your company's resources, MIS at times, needs to search for other means for capturing system resources.

When talking about computer resources, it usually comes down to CPU performance and ***"Disk I/O's."*** There is the standard fare of products that will help in both of these areas. Tools such as ***"MR NOBUFF"*** readers, high-speed relational extraction tools and hardware enhancements are just a few on the list. For MIS this may not be enough, you may need to take advantage of Other Avenues which may not have been explored before.

There are three areas that come to mind that are often overlooked: **1)** Controlling your end-user reporting environment; **2)** Taking advantage of your corporation's MIPs and disk space already accumulated in the form of PC's; **3)** And finally, getting the biggest return on investment from your MIS systems development staff.

This paper will explore these three areas of data processing in more detail so you can take advantage of some resources that you may have overlooked.

END USER COMPUTING:

Does the thought of giving an end-user reportwriter to a bunch of unsophisticated end-users conjure up visions of doom? Usually. As a DP shop you have to decide whether on the one hand it's an advantage to have end-users do their own reporting so some of your time is freed up for more complex tasks. On the other hand the users might abuse the privilege and bring the CPU to its knees which may cause harsh feelings between the users and DP. What happens then, do you take the privilege away? This of course would put the reporting load back in the hands of DP, or do you continue allowing the end-user do their own reports knowing very well that the CPU could be buried at any time. Sounds like a "Catch 22."

The Hidden Resource:

Even though we cringe at the thought of letting end-users loose with a tool like a reportwriter, the end benefits usually out weigh the penalties. You can think of your end-users as a hidden resource in a manner of speaking. They are getting the information they need themselves without putting too heavy of a burden on DP.

The Goals:

To implement an end-user tool like a reportwriter you will need to set up some criteria for selection. My recommendation for criteria is just that, a recommendation, feel free to add or subtract according to your specific needs. Some of the areas to look for are:

Ease of Use: If the product is too difficult for the user to navigate then more than likely it will be destined to failure. Look for a reportwriter that offers a lot of on-line help, menus, function keys, etc.

Flexibility: If the product is simple to use but the user can't do much with the product, then guess what? The user is coming back to you (DP) for those reports.

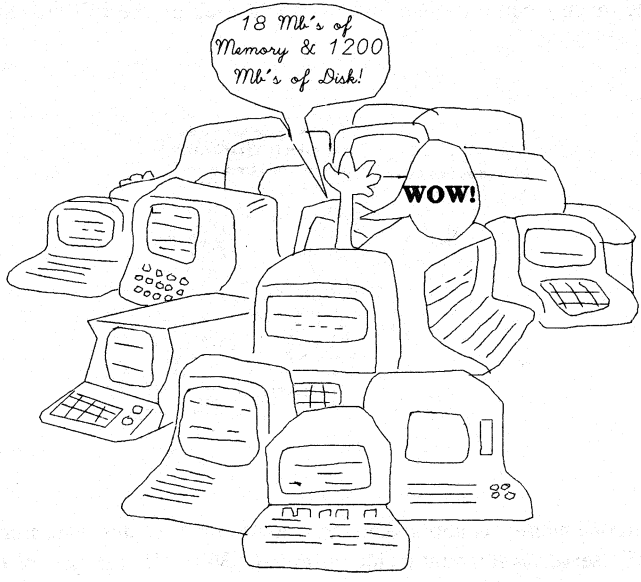
Automatic Links Into Other 3rd Party Tools: As we all know, 3rd party software vendors can offer a wealth of tools to you to take advantage of. If the reportwriter you're either using or evaluating won't interface to them, especially automatically, then you will be forced to write some interface (if the product will allow you to) or worse yet you will have to purchase one!

Control: This by far is the most important of them all. If DP has the ability to control who can run which report when and who has access to what part of the reportwriter, then you're really in control of that end-user resource.

You need to evaluate your particular environment and ask yourself some questions to determine if End User Computing is for you or your shop.

PC INTEGRATION:

In most corporations today you will find a PC or two doing specific tasks; such as word processing, spreadsheets, graphics etc.. When you talk about PC's and HP 3000 computers in the same sentence it usually means terminal emulation. That's almost an insult to the PC in a way. The thought of using this small processor and disk drives for something as mundane like being a dumb ASCII terminal is a waste. Well, all is not lost, vendors have come to the realization that, PC's are like little computers.



The Numbers:

Have you checked your PC count lately? If not, you may be surprised to learn that PC's are slowly replacing terminals for user workstations. There are two primary reasons for this, one is that the price of a PC has dropped significantly putting it into the same dollar arena as the terminals. Secondly the amount of software that is available on the PC can help the user become very productive.

Well, if all these PC's are showing up on all of these desks, then let's convert these PC's into some meaningful numbers. If your company has 30 or so PC's with 640K of memory each and 40 Mb of hard disk each this would give you 18 Mb of main memory and 1,200 Mb of disk storage!

The best part about these figures is that even though you have essentially the power of a series 70 in those PC's, each unit is an independent processor. This means each user isn't contending for CPU resources with everyone else.

The Software Catch:

With all this CPU and Disk available it almost sounds too good to be true, right? The real trick is to find some way of integrating all of those PC's to an HP 3000 to make it all a reality. There are a lot of old and new **"BUZZ"** words out there like **"Distributed Processing," "Cooperative Processing,"** or **"Client Server."** They all pretty much mean the same, run the application on the PC and share the data from the HP 3000.

To do this one needs a common data structure, a common language, along with some sort of file server that would reside on the HP 3000. In this type of environment some of the items you will want to be aware of are:

Common Data Structures: If you are using IMAGE or a combination of IMAGE with a relational type enhancement like Omnidex or Superdex on the HP 3000. Then you will want to find something close to that arrangement on the Personal Computer. If you don't you will need to have a Data Base Administrator that knows IMAGE on the 3000 and whatever on the Personal Computer. Worse yet you may have to hire a new employee that knows the data structure on the PC.

Common Development Languages: If you are planning to move an application from the HP 3000 to the PC. It will become less painful if the languages are the same. Fourth Generation Languages will give you better flexibility in this area because they have their own screen drivers etc. (View 3000 won't run on the PC). Also check with the vendor who is offering a data structure for the PC environment and make sure you have access to it from other 3rd generation languages such as COBOL.

Common User Interface: This portion sometimes become very crucial for the acceptance of a system. Let's say, for example, that you develop a system on the HP 3000 in COBOL and View. Now you replicate the system on the PC using some type of PC tool or language. Even though they accomplish the same given task, they will behave and look differently. Try to explain to the user that they have to learn two different versions of the application to get their work done. This can be very frustrating for an end user.

Data Transfer Capabilities: Batching files on groups of data up to the HP 3000 or back to the Personal Computer are fine in some cases. In your shop that idea may not be indicative of real life. To take full advantage of those PC's and your HP 3000 in a real-time interactive environment, you will need to seek a product that can pass data interactively back and forth. This may be based on demand or some preset time limits (time stamping).

One has to be careful though, if you find a vendor that can offer this type of environment (interactive or cooperative processing) don't get caught up in the flexibility of the software and kill the application. As an example, if you set up the application to continually read the data files on the HP 3000 then you would be better off just using terminals. On the other extreme the application may be interrogating the data on the HP 3000 maybe just once a month. This would be a good indication that the application wouldn't need an interactive environment to the HP 3000 data. You have to gaze into that crystal ball and really evaluate your application.

If you seek the right tools and implement a sound application then the amount of resources you will acquire from those PC's will be incredible!

APPLICATION DEVELOPMENT:

What is the most valuable and by far the most perishable resource in your Data Processing Shop? Without question **TIME**. And it's funny how every task, whether at work or home, takes valuable time. So what does one do to preserve some of that valuable resource? Work more effectively and efficiently.

Finding The Time Bandits:

Time bandits are as the name implies and they come in many forms. The people who are affected the most by these bandits are you, as well as your entire Data Processing organization. Let's try to identify these time bandits and then formulate a plan to help make your shop more efficient.

Program Backlogs: In almost every data processing shop there is a backlog. They can take on three primary forms. Programming backlogs are by far the largest and the most costly of our time bandits.

The first is new development or new request backlogs and depending on the company this backlog could be very large or hardly insignificant. If your company is suffering from a new development backlog this could be a tremendous load on data processing since the request has to be analyzed and laid out, then actually programmed. This process can take months, even years, depending on the size of the project. After it's implemented the user will always come back and say ***"this isn't what I asked for....!"***

The second type of backlog is your typical maintenance request and this could be the most dreaded and the most painful. The reason is, most of the programs currently in production are older, and written in COBOL, FORTRAN, etc. by someone who is no longer with the company or who never documented their code. Our friends, the time bandits, are very active taxing your time on these projects. Not only are they time consuming for the actual coding but the amount of time it may take to find the affected code let alone trying to figure out what the program is doing can turn into a tedious task.

The worst part of any backlog is what I refer to as **"THE HIDDEN BACKLOG."** The hidden backlog comes about when the visible backlogs become so big, the users figure by the time data processing gets around to my new request I'll be in retirement. The user may have written off the request, but it is still there in the back of their mind waiting to spring to life. Even if you start to catch up on your normal backlog and think you're ahead of the game, suddenly the user realizes that all is not lost. Now the user may feel it's a good time to submit their request once again.

The Backlog Solution: The answer to the backlog problems can be many, but be forewarned that none of the answers will come cheap to a data processing shop or company. Keep in mind, that any major purchase be it either staff, software etc. can usually be justified over time and depending on the solution you choose. Your Return on Investment could conceivably happen within a few months following the purchase.

There are two possible ways to help cut programming backlog that I'm aware of and it's entirely up to you and your environment which might fit best. The first of course is to throw more resources at the problem or quite simply add programming staff either permanently or through a consulting relationship. Adding a new member to your staff will be harder to justify then hiring a consultant. A full time person means a continual outlay of salary plus medical and retirement expenses. A consultant is more like a temporary in that you can get the help for as long or as short of a time as required.

The second alternative to your dilemma would be to implement programming aids such as code generators or 4GL's. Now I know what you're thinking, but MOST of the software vendors out there have been listening to the masses over the years. As a result, the 4GL's have become more flexible along with being performance conscious. If you take a look at the numbers, the price of a 4GL will equal the cost of a full time employee (1 year's salary) but will typically give you the programming productivity of 5 programmers! Your 4GL vendor should be able to provide you with these numbers as well as cost justification numbers to help you sell the idea to your management.

The CPU Crunch: Another time bandit that may have been overlooked in the past is CPU performance. It will affect not only the users on the production system but it can severely impact the work effectiveness of your data processing staff. Consider a programmers work cycle of using a typical editor to modify their code, save the text file, compile the program and finally test their results. You can see how this repetitive flow can turn into a very time consuming process if your current CPU is at it's limits.

The CPU Solution: Once again the two possible solutions for this particular problem could have an expensive price tag associated to it. The possible savings over a year or two could justify the expenditure. One additional note, this is not to say that these solutions are the only answer but they are the two I have arrived at.

The first option would be to look at upgrading your current CPU. This of course would require a lot of planning and financial resources. The overall benefit to all users in the company as well as data processing would be a great shot in the arm. One item to keep in mind when looking to upgrade your CPU is to gaze, once again, into your crystal ball and try to formulate the growth of the company. Then you can anticipate how large the CPU needs to be on that given year. The worst that can happen is you go to management and get the funds to upgrade a series 52 to a 70 and a year after the upgrade your computer is on it's knees again. By then I hope Your resume has been upgraded.

"... The budgets been approved for the 960!"

The second option is to consider purchasing a small 3000 like a Micro LX or GX on the Classic side. These baby 3000's pack a pretty good punch for their size and some machines cost is in the area of a well outfitted PC. The other added advantage is that they don't require any special wiring or air conditioning because they're an office environment computer. If your staff is larger than say 5 to 6 programmers, then consider maybe more than one keeping a limit of about 6 programmers per CPU. You may just want to consider going up to a 922 LX that essentially is a full blown 922 limited only by the operating system at a much more attractive price tag. Once again the 922 LX is an office environment computer not requiring any special wiring or cooling. The second alternative may not be as costly or take as much planning. I would highly recommend any data processing shop to consider this option especially if your current CPU is groaning under the load.

We have identified some of the dreaded time bandits that might be lingering in your data processing shop. I have given you some ideas to help you evolve your programming staff and DP into an efficient entity.

That's a Wrap:

Systems resources is a fairly broad term, I only hope that the information that we have mentioned here might be helpful to you in some way. At least, I hope it may have spawned some alternative ideas that will start you on your way to a more successful data processing shop.

The Myths and Facts of Performance Numbers

Husni Sayed
co-authors: Deborah Cobb, Andrew Patterson
IEM, Inc.
P.O. Box 8915
Fort Collins, CO 80525
(303) 223-6071

Introduction

The attempt to distinguish between similar hardware products can be greatly confused by the specifications published by various manufacturers. From access times to MTBF, the "numbers game" is very popular. Before comparing information that you have collected, it is important to understand what these numbers mean, how they can be calculated, and how you can compare one manufacturer's product with that of another. Some specifications that are reported can be calculated in a variety of ways—and not all manufacturers use the same method. Other specifications are so dependent upon outside factors that they are rendered meaningless without more specific information.

Speed Specifications

Almost anyone who is thinking of buying a mass storage device will begin by asking, "how fast is it?". Some speed specifications, such as Seek Time, Latency, and Average Access Time, are calculated and reported fairly consistently across different manufacturers. Others are calculated and reported with no similarity at all.

Seek Time, Latency, and Access Time (Disk Drives)

Seek time, latency, and access time are fairly commonly reported figures for disk drives, and have well established definitions. To fully understand what these are, it helps to understand how a disk is configured. Each disk has a number of concentric *tracks*, which are further divided into *sectors*. This is illustrated in Figure 1.

Seek Time is the amount of time it takes the head to position itself over a specific track. Obviously, the time it will take depends upon where the head is when it starts the seek, and how far it has to go. Therefore, seek times can be expressed in a number of ways. The *Track-to-track* seek time is the average amount of time it takes the head to move from one track to an adjacent track. *Full Stroke* seek time is how long it takes the head to move from the outermost track to the innermost track. *Average* seek time is the average amount of time the disk takes to perform a seek. This can be calculated by timing all possible seeks and taking the average. Average seek time may also be reported as the time it takes for the head to move $\frac{1}{3}$ of the way across the radius of the disk, which is generally fairly accurate.

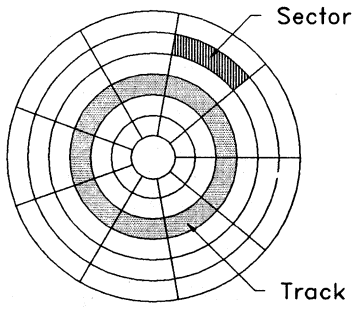


Figure 1: Disk Sectors and Tracks

Seek time usually includes *settling time*, whether or not this is indicated. When the head is moving and then stops, it will vibrate slightly before becoming stationary. The settling time is the amount of time it takes the head to stop vibrating when it comes to a stop.

Once the head is positioned over the appropriate track, it has to wait for the desired sector to come into position. *Latency* refers to how long it takes the appropriate sector to come into position beneath the head. Obviously, this is a function of the rotational speed of the disk, and how far away the appropriate sector is when the head comes into position over the track. So the average latency will decrease as the rotational speed increases.

Finally, *Access Time* is the total time it takes the disk to prepare itself to start reading or writing the requested data, which is simply the seek time plus the latency. Access time is almost always reported as an average access time (average seek time plus average latency).

Search Speed (Tape Drives)

Seek time, latency, and access times do not make sense for tape drives, which are sequential, rather than random access, devices. The search speed of a tape drive will depend upon how fast the tape drive can read (and search for a specific file) while the tape is winding forward at "search" speeds. Obviously, this will never be longer than it takes for the tape to "search" from the beginning to the end of the tape. The average search speed is generally the amount of time it would take to search to a file located in the middle of the tape.

Transfer rates

Transfer rates are perhaps the most misinterpreted figures that are reported on product data sheets, both for disk drives and tape drives. Transfer rates reported on data sheets are generally, from the application's point of view, maximum speeds, and will usually not be an accurate representation of the speed that a given user will achieve with a specific system/application. Only users with applications and equipment virtually identical to the test situation will yield the speeds shown. There are two types of transfer rates that are commonly reported: Burst and Sustained.

Burst Transfer Rate, applicable only to disk drives, is the best performance of the raw disk. This is the maximum rate at which a single sector can be read from the disk, unencumbered by overhead of the operating system, controller, or interface. It is basically a measure of how fast the head can read the data, and is related to the disk rotational speed. While the burst transfer rate can be used to compare (to some extent) the relative speeds of different devices, it will always be faster than the transfer rate achieved by the user after overhead incurred by the operating system, interface, or controller is factored in. All things being equal, a disk with a faster burst transfer rate than another will likely be faster with the user's application as well.

Sustained Transfer Rate, applicable to both tape and disk drives, is the fastest rate at which the device can transfer a continuous stream of data, unencumbered by external influences. For tape drives, this is the speed assuming that the tape is kept streaming. For disk drives, this entails transfer across sequential tracks and sectors, and is basically a factor of the Burst Transfer rate, track-to-track Seek Time, and Latency. Sustained transfer rate does not employ any random access, and assumes that whatever the information is being transferred to or from can generate or accept the information as fast as the raw device involved in the transfer.

Realistically, the transfer rate that can be expected by the user (the *Effective Transfer Rate*) will generally be much lower than either the Burst or Sustained transfer rates. However, no one reports an effective transfer rate—unless it is followed by so many disclaimers as to render it useless. An effective transfer rate is impossible to report, since it is affected by so many factors, including the raw speed of the device, the speed of the interface and computer being used, and other factors involved in the drive itself (such as internal buffering). Some of the major factors that affect the effective transfer rate are discussed below.

Factors Affecting Effective Transfer Rate

Effective transfer rates are greatly affected by many factors, among them:

1. **Device Speed**

   Of course, the speed of the device(s) being used is a great factor in transfer rate. The speed of the device includes such things as access time and transfer rate, as discussed previously. If information is being transferred from one device to another, the transfer rate will be limited to the speed of the slower of the two devices.

2. **Computer Speed**

Related to computer speed are such factors as:

(a) The interrupt response rate.

(b) The availability of DMA (Direct Memory Access) transfers.

(c) The speed of the CPU and RAM, and the amount of RAM.

Obviously, faster computers can run utilities which are used to store and restore files faster than slower computers. Faster computers also tend to have faster I/O ports.

Computer speed can also be critical when software utilities that perform data compression are being used. Data compression is highly processor intensive. With such utilities, computer speed has a very large impact on device performance.

3. **Operating System and Drivers**

Related to the operating system and device drivers that are used, the following can affect the transfer rate:

(a) Single-user vs. Multi-user systems. Obviously, a system that is supporting multiple users will be slower than a system that is solely dedicated to the task at hand.

(b) Whether or not the system uses cacheing.

(c) The structure of the device drivers, including any buffering in the device drivers.

(d) The priority system, and what priorities have been assigned to the task at hand.

4. **Interface**

There are many interfaces which can be used—HP-IB and SCSI are two common interfaces for Hewlett-Packard computers. Characteristics of the interface being used can greatly affect the transfer rate. Such characteristics include:

(a) The band width (the number of bits of data that can be transferred in parallel).

(b) The handshake speed.

(c) The length and quality of cables being used. Excessive cable length, and the use of low-quality cables, can severely affect the performance of devices on the system.

(d) The number of devices on the system.

(e) The speed of other devices on the system. This is especially important when the other devices are actually operating. However, even when other devices are not actually operating, they may participate in handshaking, which can affect the speed of other devices on the system.

5. **Size of Transfer**

There is always some overhead involved in initiating a transfer operation. This overhead is both on the computer end (processing directory entries, opening and closing files) and on the device end. And for tape devices, the writing of file marks can have a significant effect on the speed of data transfer. In all cases, a single large transfer operation will be faster than several smaller transfers, given the same amount of data:

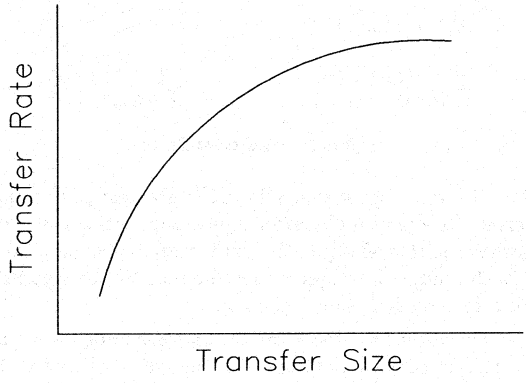


Figure 2: **How Transfer Size Affects Transfer Rate**

The size of the transfer, combined with the file system architecture, will affect transfer rates to different degrees. For example, a hierarchical file system is much more complex (entailing more "random" access) than a "flat" file system, and is therefore affected to a greater degree by fragmented transfer operations in which overhead has a more significant impact.

6. **System Activity**

System load and interface activity from other processes have a large impact on overall performance. In general, the fewer users and devices on a system, the faster the existing devices will be able to operate.

7. **Interleave Factor**

For disk devices, the interleave factor which is used when the disk is initialized may also have an affect on the speed with which information is transferred. The interleave factor specifies how sectors with sequential storage "addresses" are arranged on each track. With an interleave factor of 1, sectors with sequential storage "addresses" are located adjacent to each other (or, each sequential storage address advances 1 sector). With an interleave factor of 2, each sequential storage address advances 2 sectors, so there is 1 sector between adjacent storage addresses. This is illustrated in Figure 3.

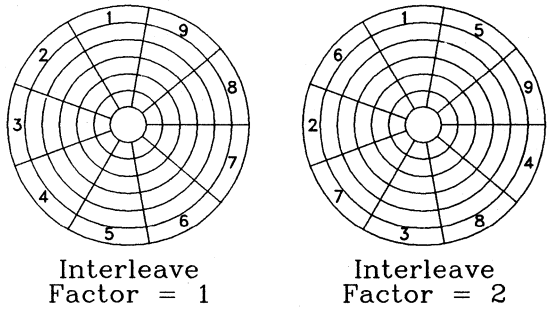


Figure 3: Interleave Factor

Finding the interleave factor that allows the greatest performance is a matter of trial and error. In many high-performance applications, the disk is being used to access large amounts of sequential data, stored in sequential storage addresses (sectors). Each read/write request must be sent by the computer and processed by the disk drive—which takes some time.

Suppose the computer has read sector 1, and has issued a request to read sector 2. If the disk has rotated past the starting point of sector 2 by the time the request has been processed, the disk will have to "wait" until sector 2 is again positioned below the head. In this case, the disk is "under-interleaved", and at least 1 latency cycle will be invoked for each read/write request.

On the other hand, the disk can also be "over-interleaved". Again, suppose the computer has read sector 1 and issued a request to read sector 2. If the disk has not yet reached the starting point of sector 2 by the time the request is processed (which could happen with an interleave factor greater than 1), a delay is again incurred while sector 2 comes into position.

Of course, it is impossible to get the disk interleaved so that the disk will always be in the desired position at the exact instant the request has been processed. To achieve maximum performance, the disk should be slightly over-interleaved: under-interleaving will always incur the longest delay, since it guarantees that the head will have to wait for almost a full rotation of the disk.

These are some of the major factors that affect system performance, but certainly not all of them. A myriad of other factors will affect system performance, such as cable quality, environmental considerations, how often a unit is cleaned, etc.

Media Capacity Specifications

Media capacity is another tricky number to interpret, especially with the introduction of the new, high-capacity helical scan tape drives. Everyone is advertising disk and tape drives with more and more capacity. Hard disks and tape drives are the two major classes of products for which capacity is reported. Hard disk capacity is fairly straightforward, while tape capacities are affected by everything from the utilities being used to store data, to the temperature in the room at the time data is stored.

Factors Affecting Disk Capacity

Disk capacities reported in data sheets will either be the formatted or the unformatted capacity. The unformatted capacity of the disk is useless information to the user. If a data sheet does not specify whether the reported capacity is formatted or unformatted, beware—formatting could use as much as 25% of the disk space, decreasing a disk with an unformatted capacity of 600 MBytes to a mere 450 MBytes.

Low-level formatting, which is performed by the manufacturer, partitions the disk into records, record headers, and record trailers. Record headers and trailers are used to identify sectors and tracks, and include error correction codes for ensuring data integrity. Formatting also places synchronization codes between records. All of this occupies space on the disk that cannot be accessed by the user.

The only other factor that affects disk capacity to a significant degree is the error rate. Errors cause sectors to be "spared", for error correction. These spared sectors cannot be used to store information, and hence reduce the capacity of the disk.

Factors Affecting Tape Capacity

Logical and Physical BOT/EOT

Tapes have both a Physical Beginning Of Tape (PBOT), and a Logical Beginning Of Tape (LBOT). Similarly, they have a Physical and Logical End Of Tape (PEOT and LEOT, respectively). This is illustrated in Figure 4.

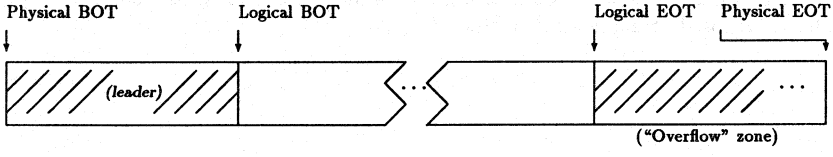


Figure 4: **Physical and Logical BOT/EOT**

LBOT is the location where users can actually begin storing information. The space preceding this (from the PBOT to the LBOT) is taken up with a tape leader. The size and contents of the tape leader will vary from one system to another, but all tapes have some space at the beginning that cannot actually be used.

Tapes also have a logical EOT, which is generally some distance from the actual physical end of the tape. This LEOT serves as a "warning" that the end of the tape is coming. When the LEOT is encountered in the middle of a store operation, the tape drive finishes what it is storing, and cleans everything up. The amount of tape that is "left" after the LEOT varies widely, from a few KBytes for 9-track tape drives, to MBytes on 8mm tape drives. In any case, the amount of tape reserved after the LEOT must be enough to at least empty the tape drive's buffer.

Tape utilities generally do not write past the LEOT, except to perform the necessary "cleanup". For drives that reserve a large amount of tape after the LEOT, this can be a significant waste.

If tape capacity is reported from logical BOT to logical EOT, it will be a fairly accurate representation of the maximum usable capacity of the tape. However, capacity measurements reported from logical BOT to physical EOT are including a potentially significant amount of tape that will probably never be used.

File Marks

For tape drives, the nature of the data being stored can have a large affect on the capacity. File marks are used on tape drives to delimit the end of one file and the beginning of the next. These file marks obviously take up space—more on some tape drives than on others. On 8mm tape drives, a single file mark typically takes up 2 MBytes of tape space. The use of "logical" file marks, which are implemented on IEM 8mm drives, can reduce this to 1024 bytes per file mark. Similarly, 4mm tape drives use roughly 128 KBytes of tape space for each file mark.

How much these file marks will affect tape capacity depends largely upon the application. For example, when performing backup operations, some backup utilities write more filemarks than others. One of the "worst-case" utilities is HP's `STORE` utility for HP 3000 computers, which appends a file mark after every file that is written. When many small files are being stored, this can severely decrease the capacity of a tape.

In all cases, for a given amount of data, it is more economical to store a few large files as opposed to many smaller files.

Computer Speed

For streaming tape drives, the "streaming" nature is important to tape capacity. Few computers will be able to keep the tape streaming at all times. This means that the tape will be starting and stopping every time the tape drive has to wait for the computer to catch up and send more information. Starting and stopping the tape creates gaps in the tape that cannot be recovered.

For example, an 8mm tape drive will use roughly 8 KBytes of tape for every stop. Assume that the computer is sending 16 KByte blocks with a full tape stop after each block. This scenario would result in an 8 KByte gap after every 16 KBytes of data—effectively reducing the tape capacity by $\frac{1}{3}$. This would reduce a 2.3 GByte tape to around 1.5 GBytes. All tape drives (4mm, 8mm, 9-track, etc.) are affected in a similar manner by tape starting/stopping.

Error Rate

Data integrity is crucial for stored information. All tape drives employ error correction techniques that correct errors "on the fly", while the tape drive continues streaming. When errors occur during the writing process, the tape drive cannot simply stop, back up, and rewrite the information. Instead, the section of tape containing the error is considered "bad", and the information is rewritten correctly. For every error that occurs, the tape capacity decreases. The amount of space required to correct errors varies depending upon the error correction techniques employed, but in all cases some tape space is required. Obviously, then, the more errors occur, the more significant the decrease in tape capacity.

Errors are going to occur in any data storage application. However, factors such as media quality, environment, and cleanliness can affect the tape capacity by increasing the error rate. Environment is an important consideration: operating the drive in an environment that does not conform to humidity and temperature requirements can stress the tape, increasing the error rate. Only data grade tapes should be used for data storage, and tapes should be replaced at the first signs of wear. Finally, tape drives should be cleaned on a regular basis to reduce errors and increase performance.

Data Compression

Data Compression allows information to be stored in a more compact form than normal, reducing the amount of space that a given amount of information occupies. Depending upon the type of information being stored, data compression techniques can compress user data to $\frac{1}{20}$ its original size (a compression ratio of 20:1). Realistically, data compression yields typical ratios of 2:1 to 5:1.

Statistical Data Compression makes use of the fact that certain characters are used more frequently than others. Typically, uncompressed data is stored such that each character occupies 8 bits. With statistical data compression, frequently used letters such as "e" occupy only 3 bits, while less commonly used letters such as "x" are stored in 13 bits. While this expands the space required for certain letters, it decreases the total number of bits required to store information, assuming that the occurrence of the most frequently used characters is high.

Repetitive Character Reduction searches for strings of at least four repeated characters in a row. Each such string is then reduced to three characters: a unique character that indicates the beginning of the group, the actual character which is repeated, and a character which indicates how many of these characters occur contiguously. If all of the repeated characters are in groups of four, the compression achieved with this method will be 25% (or 4:3). If much larger sequences of repeated characters are found, the compression ratio can be higher.

Substitutional Data Compression compresses data by substituting frequently occurring strings with a single symbol. For example, this string:

```
what you see is what you get
```

has a total of 28 characters, including the spaces between words. This can be compressed by replacing each occurrence of "`what you `" with the symbol `#`, resulting in:

```
#see is #get
```

for a total of 12 characters.

All data compression algorithms have some degree of overhead. For all of these methods, some sort of decoder, or "dictionary" must be stored with the data so that it can be decompressed when necessary. The effectiveness of data compression also varies with the "compressability" of the data. Binary data, for example, has an average compression ratio of about 1.8:1, while CAD files have an average compression ratio of more than 4:1. And some data, for example a table of random numbers, will not benefit from data compression at all. In fact, with such data, the overhead involved in the compression may be greater than the reduction achieved—resulting in the data actually expanding.

Until recently, data compression was always implemented in software. However, storage products are now beginning to appear that have data compression which is implemented in the hardware. With this new development, users should be careful. For instance, some file systems will automatically compress data before storing it to disk. If this information is then transferred to a tape drive that uses data compression, the data may expand in the move from the disk to the tape—especially if the algorithm that initially compressed the data was a good one.

Reliability Specifications

MTBF

MTBF, or *Mean Time Between Failures*, indicates the calculated average number of Power-On Hours (POH) per failure for a given product. For example, an MTBF of 30,000 POH indicates that, on the average, one failure will occur for every 30,000 hours that a unit is powered on. This would seem to indicate that MTBF is a good measure of a device's overall reliability. However, MTBFs reported by different manufacturers can vary widely, and be calculated in any number of ways. So, comparing the MTBFs of two different products may be comparing apples to oranges.

Most mechanical and electrical components have failure probabilities that change over time. These failure rates follow a pattern that is commonly called the bathtub curve. The bathtub curve, shown in the figure below, starts with a relatively high failure rate (point a) which rapidly falls to a much lower steady failure rate (point b), and then at some point, starts rising again (point c).

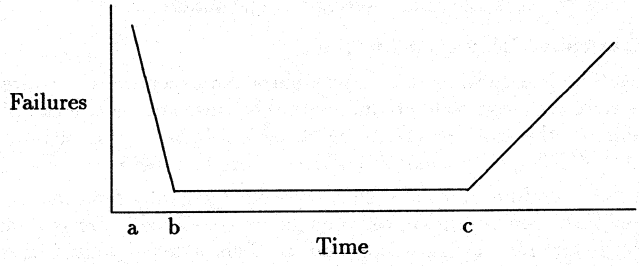


Figure 5: The "Bathtub" Curve

The portion of the curve *(a–b)* reflects high failure rates incurred by infant mortalities, caused by marginal components that fail immediately as the mechanism is stressed during its initial operation. Infant mortalities generally occur during the first month or two of operation. The portion of the curve *(b–c)* is the normal failure rate that can be expected of the mechanism. MTBF is usually measured using this portion of the curve.

Point *c* on the curve, caused by wear on mechanical and electrical components, is a good measure of the expected lifetime of the device. However, the only accurate way to determine when point *c* will be reached is to operate a large number of units until the failure rate begins to climb. Most computer peripherals would be obsolete long before point *c* could be found using this method.

Manufacturers can get a general indication of where point c will occur for a given product by examining expected life-times of individual components of a mechanism and/or by comparing the mechanism with historical data from similar mechanisms.

How MTBF is Calculated

There are many different methods used to calculate MTBF. Three of the more common methods are the Test Sample Method, the Theoretically Calculated Method, and the Demonstrated Method:

- **Test Sample Method**

  This method runs a sample population of products until failures are encountered. Data collected from these samples are then extrapolated to derive a population rate of failure. The confidence level of this method is greatly dependent upon the size of the sample and the duration of the test. Test Sample MTBF is calculated using this formula:

$$\text{MTBF} = \frac{(\#\text{ of units}) \times (\text{power on hours})}{\text{Actual failures}}$$

  So, this method could produce an MTBF of 30,000 hours either by running 1000 units for 300 hours each and encountering ten failures, or by running 60,000 units for 30 minutes each and encountering a single failure.

- **Theoretically Calculated Method**

  This method is used when it is impossible or too expensive to acquire statistical failure rates for a complete product, either because not enough sample units are available, or the cost of performing statistical tests is prohibitive. Most new product MTBF specifications are derived using this method.

  The expected failure rates of each component (generally reported by the manufacturer of the component) in the product are combined to arrive at an expected product failure rate for the entire product. This is done by dividing the product of all the component MTBFs by the sum of all the component MTBFs:

$$\text{MTBF} = \frac{\text{MTBF}_1 \times \text{MTBF}_2 \times \cdots \times \text{MTBF}_n}{\text{MTBF}_1 + \text{MTBF}_2 + \cdots + \text{MTBF}_n}$$

  In the equation, MTBF_1 is the MTBF of the first component, MTBF_2 is the MTBF of the second component, etc. (This equation can be derived using standard statistical methods which, for reasons of brevity, are not presented here).

  This method of calculating MTBF can be easily fudged by excluding components that the manufacturer "deems" unimportant. Even excluding one component can significantly alter the product MTBF. For example, if you add in the MTBF of a fan (160,000 hours) to the MTBF of a drive mechanism with an MTBF of 150,000 hours, the product MTBF drops to about 77,420 hours:

$$\text{MTBF} = \frac{160,000 \times 150,000}{160,000 + 150,000}$$

- **Demonstrated Method**

  This method is the same as the theoretically calculated method, except that actual failure data acquired from customer installations is factored into the MTBF as data becomes available. This method is probably the most accurate, as long as all failures are taken into account. Demonstrated Method MTBF can be fudged by excluding certain types of failures. Some common exclusions include units that are dead on arrival, controller or cabling problems, and failures that occur within the first 200 hours of operation.

MTBF and Duty Cycle

To further muddy the picture, different manufacturers can assume different *duty cycles* when calculating MTBF. The duty cycle is the percentage of time that the unit is actually doing something while it is powered on:

$$\text{Duty Cycle} = \frac{\text{Hours of mechanical operation}}{\text{Power On Hours}} \times 100\%$$

Consider this: MTBF is reported in Power On Hours. However, a unit is more likely to fail when it is powered on AND WORKING than when it is simply powered on and taking up desk space.

Duty cycles from 10% to 25% are commonly used, though manufacturers can use any duty cycle, from 0% (units powered on and sitting there) to 100% (units powered on and mechanically operating the entire time). Obviously, low duty cycles generate higher MTBF figures than high duty cycles. Most manufacturers do NOT report the duty cycle used to calculate MTBF.

MTTR

MTTR, or *Mean Time To Repair*, is a measure of the average amount of time it takes a *trained service technician* to repair a unit. This is actually the average amount of time it takes to disassemble the unit, replace a component, and reassemble the unit. This figure generally does not include the amount of time it takes to locate the source of the problem.

For the majority of "simple" desktop peripherals, such as disk drives and tape drives, MTTR is commonly reported as 30 minutes. If the MTTR for such a product exceeds this by much, you should question how well designed the product is, or how well trained the service technicians are. More complex products may have a higher MTTR.

Users who service their own equipment should keep in mind that MTTR is based on the abilities of a trained service technician familiar with the product. The actual effort required by an end-user to repair a product will depend greatly on the quality of the documentation and the design of the product.

For users who do NOT service their own equipment, MTTR is a basically useless figure. Those who depend upon outside sources for service are affected by such factors as the location of the service center (can the unit be dropped off for repair, or does it need to be shipped somewhere?), and the workload of the service center at the time the unit happens to need repair. For such users, it is more important to evaluate the service and support that is available from the manufacturer, and the turnaround time provided.

Life Expectancy

Some manufacturers report a life expectancy, or expected life. Even when a life expectancy is not reported, you can generally assume the life expectancy to be a few years longer than the warranty period. A product with a longer warranty period will most likely last longer than a similar product with a shorter warranty period. However, a drive with a 4-year warranty will not necessarily last 2 years longer than a drive with a 2 year warranty; nor will it necessarily last two times as long. All that can be said is that a drive with a 4-year warranty will probably last longer than one with a 2-year warranty.

Conclusion

Performance numbers are not so confusing when you understand what they represent, and what are the ways in which they can be misrepresented. While manufacturers generally do not provide false information, all are influenced to some degree by "peer pressure". One company may publish an MTBF of 100,000 for a product (with no explanation), even though that MTBF includes only a single component of the product. For other manufacturers of the same product to compete, they feel they must also report MTBF in the same manner. Reporting a more realistic MTBF of 20,000 hours, which covers all of the components in the product, may be misinterpreted by the user to mean that their drive is $\frac{1}{5}$ as reliable as the other. Since users seem to place so much emphasis on specifications, manufacturers end up always representing their products in the best light. This is rarely a completely accurate reflection of the what the user experiences.

Because of this, a product will rarely (if ever) outperform the listed specifications. Sustained transfer rate, for example, can generally be assumed to be the maximum sustained transfer rate. So if your application requires a sustained transfer rate of 200 KBytes/second, a drive that is only rated at 100 KBytes/second will not be adequate. A drive that is rated at 300 KBytes/second may or may not perform adequately, depending upon your application.

While some general information can be gleaned from performance specifications, the *only* way to accurately assess which of a variety of solutions will perform the best for your specific application is to actually try each one on your system. Most reputable manufacturers have programs whereby you can either rent equipment you are trying to evaluate, or have the use of a demonstration unit for a short period of time.

REFERENCES:

Rusnack, Michael
"*Disk Drive Reliability—Yesterday, Today, and Tommorrow*"
SuperGroup Magazine, December 1988.

Devore, Jay
"*Probability and Statistics for Engineering and the Sciences*"
Brooks/Cole, Monterey, CA.

MOTIVATING YOURSELF AND YOUR STAFF

Lynn A. Novo, CPC
Network Systems Company
10 Diana Court
Cheshire, CT 06410
203-250-1919

Motivation is in vogue again. It seems hardly a week goes by when there isn't a write up or two in the press about the importance of motivating your staff. A favorite topic these days is motivating your staff in hard economic times. Motivating your HP staff never really went out of style, but in years past, we were able to get away with paying hefty salaries as our primary motivator. Times have changed and I'm glad to see that managers are interested in learning about other ways to motivate their departments.

Surprisingly, with all the literature available, I haven't seen much that gives managers concise how-to's on how to motivate an IS staff. And what can be more demotivating as a manager than spending hours reading a management book and not finding any practical ideas that

can be put to work the very next day?

I hope to facilitate an exchange of information that will get you thinking about ways you can motivate your team of HP professionals. This is not a discussion of motivational theory. Nor will we dwell on why motivation is important to the success of your department. We'll talk about how motivation begins with you, and review some practical ways to stay upbeat as a manager. We'll compare motivation and manipulation. The carrot, stick, and soul food methods of motivating others will be outlined with specific examples of each for your HP staff. Our final topic will be a survey of common demotivators for HP professionals.

MOTIVATION BEGINS WITH YOU

Like so many things that impact our success as managers, motivation begins with us. Your personal style and attitude greatly affect how motivated your staff is. Are you a natural leader? To some it comes easily, others must work at it. Whatever your natural talent is, YOU are a great motivational influence on your staff.

Motivating Yourself As A Manager

How can you possibly motivate others unless you are motivated, alert, and enthusiastic yourself? If you demonstrate a lack of commitment or interest in your people and your job, your staff will notice. Why should they bother if their boss doesn't?

The myriad of self-help books, journals, and video tapes on the market today can bombard you with information on how to motivate yourself. At the core of this information is how to create an environment for yourself which allows natural self motivation.

* **Self evaluation:** You must first assess where you are on the motivation spectrum before you can improve. This is the hard part: stand back from yourself, be critical and honestly evaluate your present situation. What do you really think of your job? The people you work for? The people who work for you? How do these feelings impact your workplace?

Ask a colleague to critique you. You may not realize how you are coming across to your staff. Do they feel that you are a part of their team, or are you an outsider? Your motivating intentions may be thwarted by some non

verbal cue you're not even aware you're giving. In other words, the messenger may be masking the message.

It is also very effective to ask the people who report to you to evaluate you as a manager. Have you ever asked someone, "How am I doing as a manager? What could I do to improve morale in the department?" You will probably be surprised by the candor of the answers.

Strive to be outstanding in your job as manager and you will find that your staff will do the same. Self evaluation will keep you on track to meet your own goals and objectives and at the same time help you motivate your staff to greater productivity.

B

* **Early Morning Primers:** Have you ever noticed that when the day starts badly, it somehow never improves? It is really tough to be a motivating manager if you overslept, didn't get a cup of coffee, and got caught in traffic on the way into work. No matter how hard we try, there always will be some days like that. But to the extent that we can control our early morning and get the day started with organization (and even serenity if that's possible), the better motivators we will be. Here are

some practical suggestions:

- Do your best to be physically and mentally alert by the start of the work day.
- Eat breakfast.
- Schedule some exercise, even if it is only a brisk walk through the parking lot.
- Wear clothes that make you feel good.
- Don't dwell on all the negative aspects of your job or your problem employees on the way in to work.
- Do something you enjoy before arriving at work.

Some ideas:

Read the newspaper over a cup of coffee before you leave the house.

Telephone a close friend or relative for an early morning chat.

Watch TV or listen to your favorite music.

Do a small task around the house that you've been putting off.

Take the "scenic route" into work.

Everyone has an "off-day" once in awhile but as a

manager, you must force yourself to be positive. Before leaving the house, look in a mirror. What do you see? That is what will greet your staff.

* **Style of entrance:** Greet your staff by name and with a smile. Chat informally and let them know you are interested in their life outside work. Create positive energy and enthusiasm that will infect your staff throughout the day.

A programmer mentioned to me how her boss, who was usually bad-tempered, harassed, and unapproachable in the morning, came in one day with a smile on his face and stopped at each of the workstations on the way to his office to greet his staff. The atmosphere in the department that day was incredible, I was told. Solutions to problems seemed to materialize out of nowhere and new, creative ideas were generated. The whole place seemed to buzz with energy. My hope is that the manager learned a lesson and continued to make his style of entrance a motivating one.

* **Personality:** Do you know what sort of impression you make on others? Work hard to be perceived as a good

listener. Make yourself available to your staff so they know you as a person, not just the signature on their annual review. Keep your sights on your goals so that you stay motivated. A motivated manager always has a motivating effect on staff.

It is easier to keep a positive attitude if you do not dwell on negatives. This sounds simple enough, but don't we in fact think more about the projects that are overdue than those that were completed ahead of schedule? And don't we think more about the problem employee than the one that's cranking out work at a record pace? As managers, we need to try very hard to keep a perspective between the good and the bad at our workplace. Concentrating on the negatives will have a demotivating effect on us and our staff.

We surveyed HP professionals and asked them to profile the personality traits of the most motivating managers they had worked for. Here is what we were told:

Good communicator- expresses clearly what, when, and why something needs to be done.

Flexible- willing to bend when necessary. Not

afraid to admit they were wrong.

Defender- stands by IS team member and serves as "insulator" from the heat of upper management and users.

Reliable and consistent- establishes standards and keeps them; consequences of wrongdoing are predictable.

Good listener- really hears what is being said, and remembers it.

Realistic- sets achievable goals and deadlines for projects. Builds in contingency plans whenever possible.

Managers who are strong in these personality traits are able to engender trust, respect, and commitment among their staff. From this foundation, a manager will have an easier time motivating others.

THE TWO H'S OF MOTIVATING

At the core of motivating others are two words not often associated with motivation. They are what I call the two

h's of motivating: HONESTY and HUMOR. The managers we interviewed admit these concepts underlie all they do for their staff motivation. It is not so much a separate management technique as a fundamental way of life for these managers.

Honesty means calling the shots the way they are. Your staff has a right to know "the big picture" and the more often you let them see it, the more understanding they will be of the pressures on you. If your staff knows the commitment you made to get a project done, they will help you meet the goal. As IS professionals, they know that there will always be "gotya's" and unknowns, but if you are doing things that make sense, they'll go the extra mile.

Many managers overlook humor as a motivator. Humor relieves stress (you don't have any of that in your department, do you?), diffuses anger and hostility, and alleviates potential conflict.

What are some ways you can introduce humor into your workplace?

Have a "joke of the day". Each department member

(including you) brings in a cartoon or joke and posts it where everyone can see it.

Begin each meeting with a humorous quote. (The side benefit of this is that it will help your staff "shift gears" from whatever they were working on before the meeting.)

Have a "stand up" meeting for the jokester in your department to tell a funny story to the group.

Include a funny icon or cartoon on your next interoffice memo.

Any of these ways will get your staff thinking about humor as an acceptable part of your workplace. One caveat though- be careful not to make the users the brunt of all your jokes. Your overall objective as a manager is to foster good relations between departments. Make sure you're not making a laughing stock of your key user, tempting as that may be.

MOTIVATION OR MANIPULATION?

The management style you practice and your personality

will determine how people perceive your actions: do you manipulate or motivate?

Manipulation is very often the method managers use to get a project or task completed. It involves little creative thought on the manager's part but reaps short-term gains in that the work will usually get done on time.

So why motivate your staff when manipulation achieves the same results? Manipulation destroys trust, commitment and loyalty in your staff, the very foundation you are trying to build.

Workers see through a manipulative manager after a while and feel used, rejected and angry. If manipulation continues, one by one the staff members leave the department. The long term consequences outweigh any short term gain in project completion.

Your own management style and personality will determine whether your actions are perceived as manipulative or motivating. The next time you assign a project that you know will be a "stretch" for an individual on your staff, think of your motivation. Do you want Johnny to do the project because you:

are getting pressure to get it done

want to look good to your superiors

hope Johnny will fail so he realizes that he's not as good as he thinks he is

want to show him he can do more if he sets his mind to it

want to develop more skills in him to help his career

want everyone to share in the department's success.

If you and your staff share a mutual trust, you will find a free flow of information uninhibited by manipulation. It eliminates concerns way about motives. Trust enhances the effectiveness of your team and gives you a loyal, motivated group of workers.

MOTIVATING OTHERS

The instruments of motivation can be classed as the rewards, punishments, and needs satisfaction. What do you immediately think of when someone mentions motivation? Something tangible like money? Or the threat of losing your job? (I know plenty of high paid MIS Directors who have this motivator.) What about an intangible such as a promotion or greater responsibility?

As a manager, you should know all the motivating resources available to you. Some of them will depend on your company's policy and your budget, but many more will rely upon your skill as a motivating manager. The important thing is to utilize a variety of motivators and recognize that different tools will work for different individuals on your staff.

Unfortunately, there is often a great disparity between what a manager thinks is a motivator and what actually does motivate. So Step One is identifying the personality traits and "hot buttons" of the people on your staff.

Are IS People Different?

As with any field in industry, information systems attracts a wide variety of personalities.

Couger and Zawacki carried out an extensive study on computer personnel and identified several characteristics. Their findings were published in the book, Motivating and Managing Computer Personnel.

Their typical computer person displayed:

- a high growth need

- high quality performance
- high need to achieve
- low absenteeism.

If these traits are recognized within your staff members, you employ the classic data processing personality type. However it is very important not to classify all computer personnel with this personality type. As a manager you must take each of you staff members as individuals and recognize their needs, drives and motives, in order to develop personalized motivation paths.

Let's brainstorm motivators in general:

* understanding and listening by the boss
* competition with others - have a reward as a prize
* travel incentives - win a trip
* recognition from other departments or upper management
* responsibility for equipment
* time available for learning
* gifts at holiday times
* progressive career ladder within the organization
* company stock offering
* working as a team on a project
* benefit package including dependent coverage

* tuition reimbursement
* pay system based on individual performance, skills and responsiveness
* supportive coworkers
* lunch with the boss
* dress code
* job title
* management openness and honesty
* flexible work hours
* cash bonuses
* praise - open appreciation for a job well done
* personalized attention by the boss
* special tasks assignments
* private office or nice location in office
* administrative support (own secretary)
* access to classified data
* allocated parking place in lot
* Friday pizza party
* company sports teams
* fitness facility on premises or health club membership
* comp. time
* participation in conferences or user meetings
* salary increase

* support of CSL or trade publication submission
* good personal relationship with superior
* challenging assignments
* task variety
* ability to see task to conclusion with visible results
* promotion

Would everyone on your staff be motivated by all of these things? NO! The point is to know the full range of possible motivators and select the ones that are appropriate for a given individual at a given time.

CARROT, STICK AND SOUL FOOD

The instruments of motivation can be classed as the rewards, punishments, and needs satisfaction. What do you immediately think of when someone mentions motivation? Usually something tangible - more money, promotion, threat of losing your job etc.

As a manager you will be aware of the resources available to you, but it is the skill you apply in choosing them that will make these motivational vehicles effective.

Unfortunately there exists a great disparity between what

a manager thinks is a motivator to his staff and what actually does motivate them. So before administering - try to ascertain the most effective motivator. No point giving a programmer a 15% raise when what he really needs is a job title change to programmer analyst.

Successful managers will use a combination of carrot, stick, and soul food techniques to keep their staff highly productive and motivated.

Carrots (also known as "incentives, rewards or perks")
These are traditional means of motivation and are usually recognized as tangible objects such as: wage increases, cash bonuses, benefit packages, company car, trips or other incentives.

Carrot incentives are used on a short-term basis to increase enthusiasm and bring immediate benefits. They require little thought on the manager's behalf and if sources are available, are easy to administer. The draw back is that they fail to provide long-term effects. Over time incentives get taken for granted, and often must be increased or improved upon to remain effective.

HP professionals frequently earn better than average salaries because they are so specialized and scarce. Obviously the paycheck is important, but computer personnel also respond well to other incentives. A special gift of appreciation for a job well done may boost a programmer's morale and productivity far more than a cash reward.

Sticks (also recognized as threat, fear or punishment) Sticks are easiest to remember as being the opposite of what we generally think of as motivators. But the negative is just as powerful a motivator as the positives we outlined. You may not realize it, but the threat of being assigned the most boring maintenance project may in fact be what's motivating some of your staff members to get their work done on time. In the face of threat, most professionals will respond by getting the job done and on time. But what is the cost to the programmer's morale and his reduced respect for the manager?

Like carrots, sticks require little thought from the manager. Sticks achieve short-run productivity but often destroy respect, trust and commitment. They work best

when there is little loyalty on the part of the staff member in the first place.

The long term consequences of repeated stick techniques will give rise to several situations:

- the programmer may leave, resulting in higher staff turnover.
- programmer may remain but only put forth minimal effort.
- anxiety and tension may increase, causing reduced productivity and satisfaction.
- undesirable behavior is suppressed and may surface at unpredictable times.
- malicious behavior such as sabotaged equipment, security breaches, or destruction of documentation may result.

Both carrot and stick techniques are reinforcement methods, which over time can lose their impact. Both the rewards and the punishments need to be increased to remain effective. You can see how this could become impossible to maintain. A more viable and effective motivator, especially for computer personnel, is to

satisfy inner needs via the soul food method. This will be explained next.

SOUL FOOD

IS people are big soul food eaters and tend to be more motivated by inner needs satisfaction. These tend to be intangible and encompass :

- intellectual challenge
- sense of purpose
- more flexibility and responsibility
- greater freedom and recognition
- increased input into decision making

Many managers neglect the importance and beneficial effects of inner need satisfaction because it is more difficult to achieve, implement and maintain.

The motivators gathered from the early days of the computer industry have changed. The scarcity of highly specialized personnel commanded a high salary income with rapid career growth. Today the story is different. Growth is stabilizing as the industry matures, salaries are moderating and promotions come more slowly. The attitudes of IS professionals are transforming and alternative

motivators are necessary.

Over the past few months we have experienced a downturn in the economy. Corporations are finding money tight, morale low and productivity down. The traditional ways of inspiration have been curtailed and managers have become more innovative motivators.

As we have said, money is one motivator, but it is not everything. These conservative times have a beneficial effect because more attention is given to what the employee really desires or needs. Can we think of any "Motivators on a Shoestring"?

* Flex time
* Relaxed dress code
* Certificate of appreciation
* "Programmer of the Week" Award
* Pizza party or group lunch (Dutch treat)
* Joke of the Day board

One of the largest motivators for computer personnel is the nature of the job itself. Although we promised no theories on motivation, it will be helpful to refer to the work of Hackman and Oldham as adopted by Couger and

Zawacki (whom we mentioned earlier). They surveyed more than 6000 data processing professionals and identified five core job dimensions that are required for job satisfaction. They are:

1. Skill Variety - use of different skills and talents to complete a project.

2. Task variety - degree to which the job requires completion of an identifiable segment; i.e. doing a job from beginning to end with a visible outcome.

3. Task significance - extent to which the job effects other people, both in the company and the external environment.

4. Autonomy - degree to which the job provides independence and the individual's discretion in scheduling the work and determining procedures.

5. Feedback from the job - obtaining direct and clear information about performance.

There is a great need for each one of these factors to be present in computer positions. Regular assessment of each

category should provide the necessary information to keep the nature of the job as challenging and motivating to the individual as possible.

Data processing professionals, according to Hackman and Oldham, display a low social need - they do not need to interact with people or build relationships to be motivated at work. But this can be a dangerous generalization. If you have someone on your staff who is a "people person", you must recognize that building relationships and having frequent contact with a variety of people is important to that individual's job satisfaction. To motivate that type, you'll want him or her designated as the "user liaison" person, for example.

The soul food method of motivating calls on your ability to recognize the different personality types among your staff members. Allocate time to get together with your staff to jointly set realistic goals; know who they are and what they need from their job. Then do your best to give them the intangible "soul food" menu so that they will thrive in your environment.

SOME COMMON DEMOTIVATORS

Demotivators are usually the opposite to the motivators. It is useful to recognize these, however so you can clarify the motivating factors involved.

1. Physical environment is not supposed to be important to IS professionals. However, we have heard many complaints about how an unorganized work space is demotivating. Do you have lots of equipment boxes in the hallway?

2. The loss of challenge from the job or diminishing responsibilities, will cause lack of motivation.

3. Projects, assignments, and responsibilities promised in the job interview must be evident in the job. A lack of honesty quickly demotivates a new hire.

4. The inability to learn new skills and techniques will cause frustration and apathy. Keep your IS staff stimulated with new technology.

5. Insufficient feedback or only negative feedback can be demotivating. I like to "catch them doing something right" as a good One Minute Manager should.

7. Style of management affects motivation. Try to make

good hiring decisions so that a compatible personality type is found. If there is a real personality clash between you and one of your staff members, you may find that the only motivation that works is stick (threat) methods. If this results in the loss of that staff member, it may be for the best.

8. Unfair or unevenly applied rewards kill motivation. Be consistent with all your staff members, so that no one is perceived as your "pet".

CONCLUSION

We have just scratched the surface on this topic but hopefully it will help each of us think about creative ways to motivate our staff. Motivation starts with us, so the way you start each work day and the attitudes you bring into your workplace are important. When you try to motivate others, remember your whole menu of motivating methods: carrots, sticks, and soul food. Recognize that individuals on your staff will need different things to be motivated. I hope we've shared a wide variety of motivators so that you can keep a productive and motivated HP staff.

TITLE: HP Employment Plans In The 1990's

AUTHOR: Jerry Lindsey
CompuSearch of Chatham County
5 Cole Park Plaza
Chapel Hill, NC 27514
(919) 942-5327

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 5109

You Want That Ad Hoc Report When? Then Do It Yourself!

Brian T. Nakagawa
Stars To Go, Inc.
4944 E. Clinton, Suite 103
Fresno, CA 93727
(209) 454-0800

Abstract: After several failed attempts with "user friendly" report writers we implemented one which is currently being used by senior managers and staff. Our first attempt was with an IBM 4381 using ADR's PC DATACOM. This package required the data be downloaded to a PC then reported via the supplied report writer or PC spreadsheets. It failed since it took too much time to download the data. The next attempt was to teach users how to write QUIZ code on an HP 3000 series 950 which failed because of the need to memorize syntax and correctly type in field names. We finally succeeded in meeting the user's requests with QuizPLUS. After 15 minutes of training, the users can create their reports in 5 minutes by selecting menu items. This paper will detail our experience with the three report writers, the factors which contributed to the failure of the first two attempts then discuss the factors which made our current system successful.

Company Background

Stars To Go currently services 1200 video rental centers in convenience stores, supermarkets, and hotels nationwide. MIS is based in Fresno, California, corporate offices are in Los Angeles, a distribution center is in Memphis, and 4 district offices are in Minneapolis, Dayton, Pittsburgh, and Washington D.C.

Each rental location has an IBM compatible microcomputer which uploads data on a daily basis. The host computers in Fresno consist of 3 HP Micro 3000's and one HP 3000 series 950. Connected to the 950 via leased lines are 5 user terminals and 2 remote printers in Memphis and 20 terminals and 4 remote printers in Los Angeles. An HP 3000 series 70 is in the distribution center which drives an automated video tape sorter. The 70 is linked to the 950 via a leased line.

Our MIS staff consists of 15 people; 2 in administration, 7 in operations, 5 in programming, and 1 in technical consultation. Of the 5 in programming, one is a programmer who averages 2 days per week on ad hoc projects. As of this writing, the support for

simple ad hoc reports is handled by the author, the Director of MIS, which takes less than 2 hours per week.

Since the company's start in 1984, there has always been strong upper management support for Electronic Information Systems. By the end of 1985, 200 video rental locations had computers and the host system was an HP 3000 series 42. By the end of 1987, there were 5500 video rental locations with computers, two HP 3000 series 70's, one HP 3000 series 58, four HP 3000 series 37's, and one IBM 4381. Supporting these systems was an MIS staff of 50.

The microcomputer in the store in 1985 represented a marketing edge for us since it decreased the average time to rent a movie from 3 minutes to 1 minute. The communications network we set up allowed each system to upload its transactions on a daily basis. This helped our cash flow because we could bill our customers on a weekly basis.

User and MIS Requirements

In MIS, one programmer is allocated for approximately 2 days per week to handle ad hoc requests. During the day, this person handles support and also works on long term projects. The procedure we've set up for ad hoc reports requires at minimum a user approved report layout (which can be handwritten) and code which conforms to our standards.

Many of the ad hoc reports do not require the expertise of an experienced programmer. We have experimented with training operators and data entry staff to handle some of these assignments, however, the report writing tools require more expertise than they have demonstrated.

Therefore, the MIS requirement for the "user-friendly" report writer is that a non-technical person can be trained in a short period of time to write simple reports. It must also be easy to maintain by someone with less technical expertise than the programmers.

The users who requested a report writer consisted of only two people. However, because they are the Vice Presidents of Field Operations/Distribution and

Corporate Development, they represent over 75% of the staff in the company. Both of these users are comfortable with spreadsheets and word processing. One of the VPs sometimes takes a laptop with him while traveling. They want the ability to generate the reports they need, when they need it, but not by becoming a programming expert.

They demand this flexibility because of the nature of ad hoc reports which is usually to address a current problem. The ad hoc report MIS generates frequently is what was requested but not what was really needed. Other times one report raises additional questions which require more ad hoc reports. There are also times when the programming staff is not available due to other commitments.

Selection

The selection of a "user-friendly" report writer is usually limited by the hardware that is being used. It is unlikely that the availability of a "good" report writer on another platform will justify the move to another computer. The same applies to the database that is used. Therefore, the following report writer selections were not a result of an extensive evaluation of all report writers but in all cases what happened to be available on the hardware and database currently used.

Attempt #1 - ADR's PC DATACOM on an IBM 4381

Note: Applied Data Research, Inc. in 1988 was merged into Computer Associates.

The ability for this system to download the appropriate data to a PC where the data would be reported via PC DATACOM's report writer or other common packages such as dBASE or Lotus 1-2-3 was one of the main reasons for the selection of this package. The rationale was that the mainframe would be the repository of large amounts of data, the report writer would extract the appropriate data and download it to the PC, then the PC would report the data using tools already familiar to the user.

We anticipated that all district offices would be able to get timely video tape history and daily sales

information which could then be summarized to provide timely weekly and monthly reports without having to use the resources of the mainframe. Batch files on the PC would automatically download the data and report the information prior to 8 am local time at each district office.

About 1 month's worth of development by 1 programmer and a writer was needed to code and document the mainframe and PC software. We piloted the system in the office of the Vice President for Field Operations in Memphis who is in charge of all of the district offices. After installing the software and a 9600 baud modem, one hour of training was given to a secretary. See Figures 1 to 6 which are sample screens of the interaction with the system.

The test of this system was stopped after only two weeks. Although the reporting of the data was timely, the extract and download of 5000 records took over 2 hours. This was unacceptable. This was a function of the protocol used between the PC software and the host software. It also took over 50 keystrokes and 15 screens to get the current status of a video tape. This was also unacceptable. The project was stopped at that time.

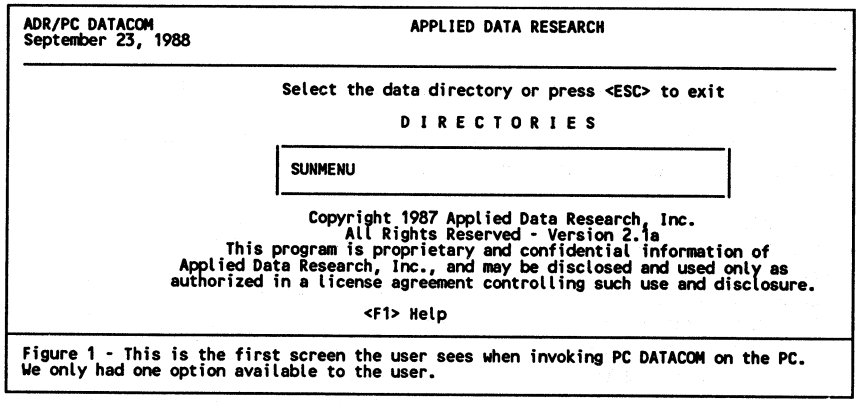


Figure 1 - This is the first screen the user sees when invoking PC DATACOM on the PC. We only had one option available to the user.

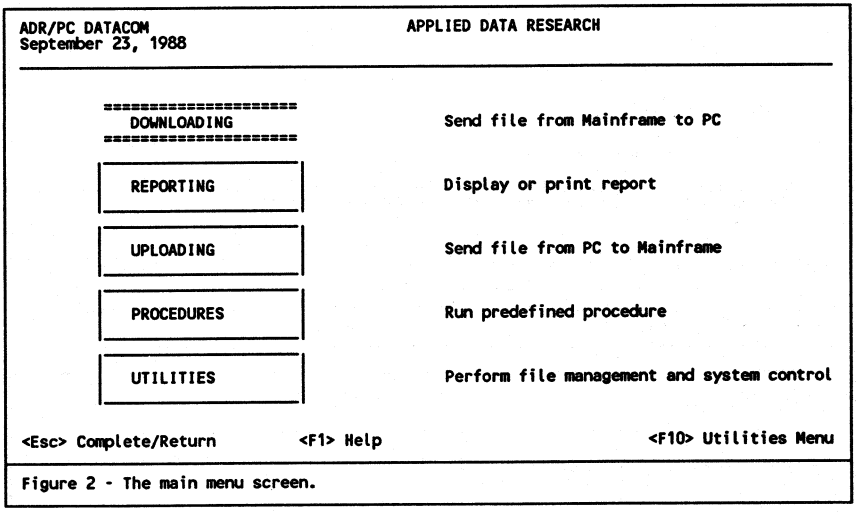


Figure 2 - The main menu screen.

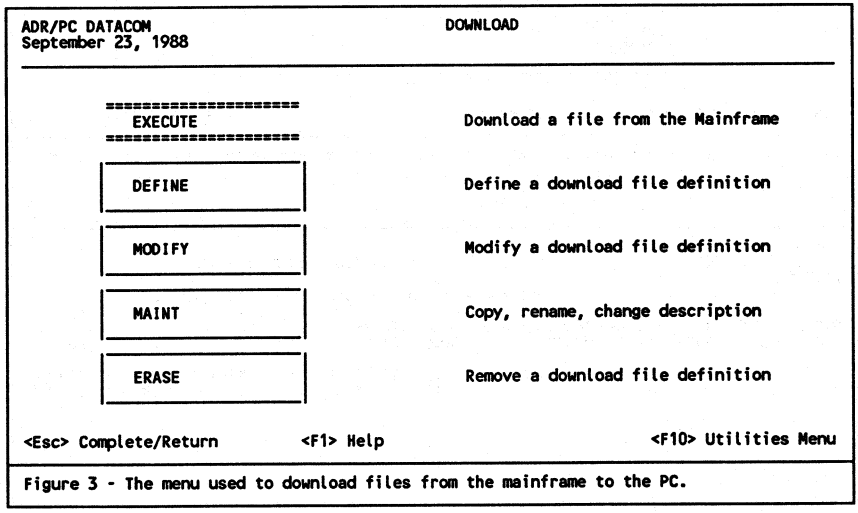


Figure 3 - The menu used to download files from the mainframe to the PC.

```
ADR/PC DATACOM                          DOWNLOAD - EXECUTE
September 23, 1988

  PC File:    FOPE520F

        Number of records found  . . . .  5

  If you do not wish to download all of the Found records, indicate
  the number of records to download above before pressing <ENTER>.

<Esc> Complete/Return        <F1> Help                  <F10> Utilities Menu
```

Figure 4 - The mainframe would show the number of records extracted for this download and ask the user to verify that all were to be downloaded.

```
ADR/PC DATACOM                          REPORTING- EXECUTE
September 23, 1988

  Select the report to execute

                    R E P O R T   D E F I N I T I O N S

                    STANDARD
                    FOPR500F
                    FOPR520F

<Esc> Complete/Return        <F1> Help                  <F10> Utilities Menu
```

Figure 5 - After downloading the data, the user needed to choose the report definition.

```
October 12, 1988                    STARS TO GO 'SUN' USER NETWORK
PAGE # 1
                              Tape and VCP location and status information

STORE  ITEM   ITEM  TRANS  TRANS  TRANS  TRANS  TRANS
  #     #     TYPE  TYPE   MONTH  DAY    YEAR   TIME      NAME              ST.  PHONE

2307  31119  P     MI     9      6      1988   20137656  XYZ STORE #1190   TX  9155551234
2307  31119  P     MI     9      4      1988   19536539  XYZ STORE #1190   TX  9155551234
```

Figure 6 - A sample report.

Attempt #2 - QUIZ

The development of the software on the HP 3000's has always been done in Powerhouse which is a fourth generation development system by the Cognos Corporation. QUIZ is the name of the report writer.

QUIZ was selected for the next attempt at providing the end users with a "user-friendly" report writer because it was already available to the programmers. Two non-technical MIS staff people had positive experiences with writing simple reports so we thought that this could be transferred to the users.

It was anticipated that less than 15 commands in QUIZ and MPE would meet most of the needs of the users. Help and examples of these commands were made available on-line. It was also anticipated that in the beginning, the users would be talked through the creation of a QUIZ report. Remember, that the two targeted users were based in Los Angeles and Memphis and would have to learn QUIZ over the phone. The plan was after a few "hand-holding" sessions, the users would start experimenting with QUIZ on their own.

Figures 7 to 11 show some of the screens the users accessed in writing their own reports.

```
                Common Commands for Online Users

  >:HELP        - This screen of common commands for online users.
  >:HELP TOPICS - Lists topics that have help messages.
  >:LISTF       - Lists only the names of your saved program.
  >:DIR         - Lists names and brief descriptions of your saved programs.
  >:PURGE XXXXX - Removes saved program XXX.
  >:STANDARD    - Lists the standard format for QUIZ programs.
  >:EDIT        - EDITOR lets you delete lines from your saved programs.

  >USE XXXXXXX  - Uses your saved programs.  Type GO to run the program.
  >[SHIFT] [Prt Sc] - Prints the information on your screen.  While
                      holding down the [SHIFT] key, press the [Prt Sc]
                      key.  [Prt Sc] means [Print Screen].
  >EXIT         - Exit from QUIZ and automatically log off the mainframe.

Q U I Z   (5.03.09)      Level 90
Copyright 1989 COGNOS INCORPORATED

>
```

Figure 7 - This is the first screen the user sees when invoking QUIZ.

```
> :dir
          * DIRECTORY OF JERRY *
-----Printing CHAIN.JERRY.ONLINE
-----Printing DAILY.JERRY.ONLINE
-----Printing DAILY2.JERRY.ONLINE
-----Printing MEMO0528.JERRY.ONLINE
-----Printing MEMOLOG.JERRY.ONLINE
-----Printing MONTHLY.JERRY.ONLINE
-----Printing PRICES2.JERRY.ONLINE
-----Printing STORLIST.JERRY.ONLINE
-----Printing TEMP.JERRY.ONLINE
-----Printing WEEKLY.JERRY.ONLINE
-----Printing WEEKLY2.JERRY.ONLINE
;CHAIN - YESTERDAY'S SALES FOR KASH N' KARRY STORES

;DAILY - DAILY SALES - CAN CHOOSE STORES

;DAILY - DAILY SALES - CAN CHOOSE STORES

;MEMOLOG - YESTERDAY'S TRANSACTIONS FOR ONE STORE

;MONTHLY - MONTHLY SALES FOR KASH N' KARRY

;PRICES2 - PRICES FOR YESTERDAY'S TRANSACTIONS
;          CAN CHOOSE STORE

;STORLIST - ALL KASH N' KARRY STORES

;WEEKLY - WEEKLY SALES - CAN CHOOSE STORES

;WEEKLY2 - LAST 4 WEEKS SALES FOR ALL KASH N' KARRY STORES
```

Figure 8 - A UDC was created to print the first three lines of the QUIZ code file which should have the name of the file and a short description.

```
> :help topics

FILENAME

TOPICSH

                HELP Topics for Online Users

     To get more information on the following topics,
type :HELP XXXXXXXX, where XXXXXXXX is one of the topics.
For example, :HELP DATASET will give you more information
on the data files you can access.

DATASET - Lists datasets online users will commonly access.
PRINT   - How to print the information on the screen.

     More HELP topics will be made available as the need arises.
```

Figure 9 - Here is a sample help screen.

```
> USE DAILY
> ;DAILY - DAILY SALES - CAN CHOOSE STORES
> ACCESS DAILY-SALES
> CHOOSE STORE-NO PARM
> SORT ON STORE-NO ON POST-DATE D
> SET REPORT LIMIT 100
> REPORT STORE-NO POST-DATE COURTESY-COUNT COURTESY-DOL TAPE-OUT-COUNT TOTAL-INV-DOL
> GO
```

Figure 10 - To run a report that has already been written, the user types "USE <FILENAME>" and the code will be displayed. To execute the code the use types "GO".

```
Store No: 0715
Store No:

08/29/90                  STG INVENTORY MANAGEMENT SYSTEM                 PAGE    1

  Store    Post      Courtesy  Courtesy  Tape    Total
   No      Date       Count      Dol     Out     Inv
                                         Count   Dol

  0715   06/26/90       0         .00     43    100.64
  0715   06/25/90       0         .00     43    115.58
  0715   06/24/90       C         .00     75    177.35
  0715   06/23/90       0         .00    118    281.95
  0715   06/22/90       0         .00     61    151.40
  0715   06/21/90       0         .00     46     95.60
  0715   06/20/90       0         .00     67    168.35
?

Records selected:   7
Records sorted:     7
Lines printed:      10
Pages printed:      1
```

Figure 11 - The code prompts for a store number then displays some sales information.

After a month and about 10 ad hoc reports it became obvious that the two users were not going to learn QUIZ and that I would always be talking them through the creation of a report. Since they were not taught how to use an editor, when a mistake was made in typing in a line of QUIZ code, it would require retyping the entire line. The biggest problem was with field names that were complicated such as TAPE-LATE-PD-DOL which is the dollar amount collected for late fees on video rentals.

One of the users made several attempts to write his own programs but would always end up calling me and having to be walked through. This was not his fault because he would write QUIZ code an average of 30 minutes per week. It is unreasonable to expect this person to become proficient in QUIZ coding.

It was decided that using QUIZ was still a way to relieve the ad hoc programmer of these simpler tasks. The MIS Director, the Development Manager, and the Operations Manager could help walk users through coding the reports, however, the number of users we supported would be limited.

Attempt #3- QuizPLUS

One of the reasons for choosing QUIZ was that the license was already paid for and the maintenance costs were already budgeted. After seeing a demonstration of QuizPLUS we were convinced that an evaluation of the product was necessary. We would worry about convincing management to spend the money on QuizPLUS after we had shown how easy it would be to write their own reports. Remember that the two users requesting the report writer would also make the decision to purchase it.

Because there were already about 10 user ad hoc reports that had been written in QUIZ, we had a good idea what the users would require. We spent about 1 week doing the necessary system administration activities to set up the program and in less than 1 day had the existing 10 reports converted from QUIZ to QuizPLUS. The time consuming parts of the system administration function was the defining of the 11 common ACCESS statements, and deciding the name of the fields that would be made available to the users.

We purposely did not send the manual to the first two users. Only one page (see figure 13) that described the datasets and the fields was sent. With the first user, in 15 minutes over the phone, he felt ready to write programs on his own. The second user had already created 4 reports before he called me for help on the 5th report. Two other users have since requested to be given access to QuizPLUS and in both cases after only 15 minutes of training feel ready to try it on their own.

See Figure 12 and Figures 14 to 16 for sample screens of interaction with QuizPLUS. Also, Figures 17 to 19 are sample screens the system administrator uses to set up QuizPLUS.

The senior managers were convinced that QuizPLUS would give them the flexibility they wanted to write their own reports and gave their approval to purchase it.

```
MODE:  ACTION:                    QuizPLUS                    09/07/90
Version 5.09D                  Reportwriter Menu

 01  Create a QuizPLUS Report     Specify report content and layout with
                                  options to run and/or catalogue the
                                  source generated by QuizPLUS.

 02  Edit a QuizPLUS Report       Revise, copy or delete a catalogued
                                  QuizPLUS report with options to run and
                                  recatalogue the report.

 03  Run a Catalogued Report      Select and run a catalogued QuizPLUS
                                  report or a Quiz report created outside
                                  QuizPLUS and catalogued using Option

 04  Catalogue a Quiz Report      Catalogue or uncatalogue a Quiz report
                                  created outside QuizPLUS referencing
                                  the source and the compiled names.

                  Enter 'E' to exit from QuizPLUS
```

Figure 12 - QuizPLUS's main menu.

```
MODE:e ACTION:          Revise, Copy or Delete a QuizPLUS Report     QuizPLUS

    Reports
 01 DAILY          06 WEEKLY         11 PINKC          16
 02 DAILY2         07 WEEKLY2        12                17
 03 MEMOLOG        08 PRICES2        13                18
 04 MONTHLY        09 MTHTODA        14                19
 05 STORLST        10 BRIAN1         15                20

            Report Name   DAILY
         21 Source Name   DAILY
         22 Compile Name  DAILYC
      REPORT DESCRIPTION:

      23 DAILY SALES-COURTESY # / COURTESY $
      24 CAN CHOOSE STORE
      25
      26

      27 Action

Enter (R)evise (C)opy (D)elete or (E)xit   (R)
```

Figure 14 - This screen allows the user to choose from reports that have already been written to be revised. The reports are always chosen by number rather than name.

QUIZPLUS Views (QPSVIEW.WKS)
June 14, 1990

Purpose	View Name	Pick one store at a time	STORE-NO	CLIENT-STORE	STORE-NAME	CHAIN-ID	OPEN-DATE	STORE STATUS	CURRENT-MONTH	LAST-MONTH	2-MONTHS-AGO	3-MONTHS-AGO	4-MONTHS-AGO	5-MONTHS-AGO	6-MONTHS-AGO	POST-DATE	WEEK-ENDING	TOTAL-INV-#	TOTAL-INV-$	TOTAL-TAX-$	TOTAL-DEPST-$	TAPE-RENTED-#	TAPE-RENTED-$	LATE-TAPE-#	LATE-TAPE-$	WV-TAPE-#	WV-TAPE-$	PROMO-#	PROMO-$	COURTESY-#	COURTESY-$	VCP-RENTED-#	VCP-RENTED-$	LATE-VCP-#	LATE-VCP-$	WV-VCP-#	WV-VCP-$	COMM-ERR-#	TRANS-DATE	TRANS-TIME	CUST-NO	COURTESY-USED	PROMO-USED	PRICE-RENTED	STRTK-PR-CODE	MVMGT-PR-CODE	TAPE-NO	TITLE-NO	TITLE-NAME	CATEGORY	RELEASE DATE	TAPE-STATUS
Store Information	STORES1	Yes	x	x	x	x	x	x																																												
	STORES2		x	x	x	x	x	x																																												
Last 6 months sales and current month to date	MNTHLY1	Yes	x	x	x	x	x	x	x	x	x	x	x	x	x																																							
	MNTHLY2		x	x	x	x	x	x	x	x	x	x	x	x	x																																						
Last 24 weeks sales	WEEKLY1	Yes	x	x	x	x	x	x									x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																
	WEEKLY2		x	x	x	x	x	x									x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																
Last 90 days sales	DAILY1	Yes	x	x	x	x	x	x								x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																
	DAILY2		x	x	x	x	x	x								x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																
Yesterday's sales	YSTDAY1	Yes	x	x	x	x	x	x																													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	YSTDAY2		x	x	x	x	x	x																													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Tape/Title information	TAPES1	Yes	x	x	x	x	x	x																																										x	x	x	x	x	x	x

Figure 13 - This is the only documentation that has been sent to each user. It shows the views (datasets) and fields that can be reported.

```
MODE:e ACTION:          View DAILY1        Name DAILY                  QuizPLUS

 Fields
 01 CHAIN-ID           04 COURTESY-#       07 POST-DATE        10 STORE-NAME
 02 CLIENT-STORE       05 COURTESY-$       08 PROMO-#          11 STORE-NO
 03 COMM-ERR-#         06 OPEN-DATE        09 PROMO-$          12 STORE_STATUS
_____________________________ Reported fields _____________________________

                   6  STORE-NO      4  POST-DATE     6  PROMO-#       4
  PROMO-$          7  TOTAL-INV-#   4  TOTAL-INV-$   7  COURTESY-#    4
  COURTESY-$       7  COURTESY-$    7  COURTESY-$    7  COURTESY-$    7
                                            Total Report Width:   82
_______________________________ Sort Fields _______________________________

   CLIENT-STORE  A N        POST-DATE     D N

____________________________ Selection Criteria ___________________________

Please pick the number beside the field. - The old value was CLIENT-STORE
```

Figure 15 - This screen allows the user to modify the contents of the report. All field names are chosen by numbers.

```
Store No: 0715
Store No:

09/07/90                STG INVENTORY MANAGEMENT SYSTEM                PAGE  1
              DAILY SALES-COURTESY # / COURTESY $ - DAILY - JERRY2

Client  Store   Post    Promo   Promo    Total     Total    Courtesy
Store    No     Date      #       $      # of      Gross       #
  No                                   Invoices      $

  Courtesy  Courtesy  Courtesy  Courtesy
     $         $         $         $

715      0715  06/26/90     7    20.93     35      100.64       0
     .00       .00       .00       .00
         0715  06/25/90     3     7.97     38      115.58       0
     .00       .00       .00       .00
         0715  06/24/90    14    35.86     61      177.35       0
     .00       .00       .00       .00
?
```

Figure 16 - A user generated report.

```
MODE:  ACTION:                    QuizPLUS                        09/07/90
Version 5.09D                    Setup Menu
__________________________________________________________________________

                        01   Views
                        02   Defines
                        03   Report Catalogue
                        04   Usages
                        05   Element Parameters
                        06   Footing and Heading Parameters
                        07   Set Parameters
                        08   Format Parameters
                        09   Elements
                        10   Load Additional Dictionary
```

Figure 17 - QuizPLUS's system administrator main menu.

```
MODE:F ACTION:                    Views                          QuizPLUS

VIEW NO:  05      01 VIEW NAME: WEEKLY1

02 SET DICT QPSDICTC.PUB.ONLINE                  03 SET DBMODE  5

04 ACCESS STORE-MAST LINK TO WEEKLY-SALES


05 CHOOSE STORE-NO PARM

06 SELECT WEEKLY-SALES IF RECORD-TYPE = "RP"

        Element                    Short Name     Type   Size   Define
07 CHAIN-ID                        CHAIN-ID        C       2      N
08 CLIENT-STORE-NO                 CLIENT-STORE    C       6      N
09 COMM-ERRORS                     COMM-ERR-#      N       4      N
10 COURTESY-COUNT                  COURTESY-#      N       4      N
11 COURTESY-DOL                    COURTESY-$      N       7      N
12 OPEN-DATE                       OPEN-DATE       D       6      N
13 PERIOD-END-DATE                 WEEK-ENDING     D       6      N
14 PROMO-COUNT                     PROMO-#         N       4      N
```

Figure 18 - Screen for modifying views by system administrator. Notice QUIZ code on lines 04, 05, and 06.

```
MODE:F ACTION:                    Defines                        QuizPLUS


01 View#:                  12    TAPES2
02 Define Name:            MVMGT-PR-CODE
03 Usage Name:
04 Heading:                Movie Mgmt^Price Code
05 Picture:
06 Last Seq#:               1


07  1 PRIORITY OF TITLE-MAST [2:1]
08  2
09  3
10  4
11  5
```

Figure 19 - Screen for modifying defines by system administrator.

Purchase Justification

The following reasons were given by management and MIS for the purchase of QuizPLUS.

1. Senior management, who interact with customers on a daily basis, can have up to the minute information on customers.
2. Senior management is able to pull up the information while traveling via laptop PCs.
3. Reports are done more efficiently by the user since it does not have to go to a programmer.

4. Programmers do not waste their expertise on "easy" reports.

Elements of a "User-Friendly" Report Writer

Although our exposure to report writers was only with three of them, the following features made the use of QuizPLUS successful.

1. Does not require keying in of syntax.
2. Ability to have summary and detail reporting.
3. Ability for system administrator to easily setup, maintain, and modify.
4. Selection of datasets and fields can be limited.
5. Dataset names and field names can be changed so they make sense to users.
6. Simple queries can be written and reported within 5 minutes of logon.
7. Initial training of less than 15 minutes.
8. Support can be by someone with less programming experience than a programmer.

Future Plans

As of this writing, although the evaluation of QuizPLUS resulted in its purchase and the two users are using it on production data, it is considered to be in the initial phase of development since it needs to be turned over to Operations for ongoing support.

We have been asked by the Vice President for Finance and Accounting to interface QuizPLUS with our accounting package on the HP 3000. Also, we are considering giving each District Office access to QuizPLUS and each account executive a laptop PC and access to QuizPLUS.

Paper #5111

The 10 Best Kept Secrets of Managing Technical People

Jason M. Goertz

Mattedor Computer Services, Inc.
5508 183rd Ave E.
Sumner, WA 98390
(206)-863-3960 Fax: (206)-863-3052

Introduction

In many data processing shops, management of personnel can constitute a real problem. It is now generally accepted that a company's people are its most valuable assets, not its capital goods or cash on hand. Proper, thoughtful management of these people is key to getting the most of this asset, just as proper management of the computer system is necessary for maximum utilization of that resource.

But quality management of these technical people is often not up to the level needed to really maximize this human resource. This situation can occur for a number of reasons. Generally, they boil down to one simple cause: inadequate training for the managers.

Over the last 11 years, I have had the chance to work with many different customers in different industries, first as an SE for HP and now as an independent consultant. In the middle of these two career stations, I worked as a manager with a department of 40 people. I had a rather severe indoctrination to management at that time, and had to learn a great deal very quickly. In recent months, I've managed a couple of people on several different projects.

I do not claim to be an expert at management and there are multitudes of people who have more experience than I do, but I have had to work shoulder-to-shoulder with a great many disgruntled employees over these years, and I have observed some management practices that work and a large number that don't. It is these observations which I hope to share with you.

What This Paper is Not

The title of this paper might have set some expectations about the content of this paper. To be honest, the title of this tome is a come-on, as this paper really does **not** contain a how-to guide for managing technical people. It contains some **fundamental** suggestions

about managing human beings. These apply to all people; technical data processors as well as clerks in an accounting department.

This really points out something that could be considered

Secret #0

Managing Technical People Is No Different Than Managing Anyone Else

The implication here is that there is a myth that technical people deserve special treatment in order to get good work from them. Nothing could be further from the truth. I will be the first to admit that people with large egos are attracted to this business, and a person with a healthy ego can be a management challenge, but I don't believe that they need to be treated any more differently than any other employee.

We will not be talking about project management either, although these ideas can certainly be applied to the human aspects of a project.

As I just mentioned, this paper is not a how-to guide to managing people. Dealing with human beings is far too complicated an affair with too many variables to even think about taking every situation into account. Instead, I'd like to talk about some general **principles** in the management of people. The best analogy I can think of is that of Obiwan Kenobi and Luke Skywalker. When Obiwan (and later Yoda) went about teaching Luke to be a Jedi Knight, they didn't teach him how to fight with a light sabre per se, but instead taught him the principle of using the Force for strength. This would then give Luke strength in future battles involving all sorts of weapons. I won't for a minute pretend I am as wise as Ben or Yoda, but I hope you get the idea of what I'm trying to accomplish. Learn some very fundamental principles, and these can be applied to all situations. Trying to learn or memorize a laundry list of how-to rules for managing people will eventually fail, since this method cannot cover every eventuality.

Intended Audience

This paper is intended for managers of all types, but most likely the managers reading this are managing people whose jobs are of a technical nature. These managers will fall into two categories, each with their own strong and weak points. The first type of technical manager are those people who themselves have a technical background, and have been promoted from a technical position to management. I am one of these. The main disadvantage to this type of manager is that they have to unlearn producing results themselves and learn to produce results through their people. Very often, these people don't make good managers because the skill sets needed for managing people (right

brained) and doing technical tasks (left brained) are very opposite. It is an uncommon individual who is equally adept at both types of skills.

The second type of managers are those whose background is not technical, but find themselves managing some group of an MIS department. MIS and DP departments are often managed by the CEO or CFO, and these people often find themselves managing technoids. These people many times have technical backgrounds in other fields, but are often professional managers. They perceive they have a disadvantage in that the technical subordinates can blow things past them and they won't understand or be able to respond. I will address some of these problems later.

Final Note

One final note before we begin in earnest. As usual, my typical caveat is in force. Take any of this as my opinion. I am convinced that virtually every management problem that exists can be traced to the lack of one or the other of the 10 ideas I am about to present, and perhaps a few that I forget to include.

Some Definitions

Before going on, I think it is a good idea to define exactly what I mean by "managing". A model that Stephen Covey presents in his seminar "The Art of Executive Excellence" is to define three roles in a business (or any system really). The first role is that of the producer. Producers are called "individual contributors" by HP and by other euphemisms by others, such as "grunts", "worker bees" etc. It is the producer that produces the end result of the company or organization.

The second role is the manager. This person is the one who makes sure the producer is producing. This person sets goals, makes sure the goals are met, and evaluates the results the producers produce.

The third role is that of the leader. The leader sets the direction that the organization will take. The manager then sets goals within this framework. Leaders can be divided into two categories. The first is the visionary leader. This person creates the vision and sees a large goal for his or her organization. Steve Jobs is this type of leader. He had (and still has) a vision that computers will be as ubiquitous as telephones. His personal interest in the Macintosh was a direct result of this vision.

The other type of leader is the team builder. This leader gathers a team of people together and gets them excited about a vision. They then go off and execute the goal. President Reagan was this type of leader. He assembled a team, set some broad guidelines, and let them go. He has been criticized for being a hands off president. The

problem, in the context of this model, is that his critics wanted him to be a manager, he wanted to be (and was) a leader. You decide.

A story or joke best illustrates the difference in the roles. Imagine a group traveling through the jungle. The producers are the people who wield the machetes and hack their way through the underbrush. The managers stand behind them and decide how often the machete wielders have to stop and rest, and how often to sharpen the machetes. They evaluate the machete wielders and rate their effectiveness by perhaps measuring the amount of brush removed in a unit of time. The leaders, on the other hand, climb a tree and see what route through the jungle they should take, and determine whether they are even in the right jungle.

I have a favorite definition of what a manager is or does. This was first put forth by former CEO of Herman Miller Max O. De Pree in his book *Servant Leadership*. Simply put, a manager is a servant who sets a goal for an employee, describes the goal and points them to it, then removes the obstacles that the employees face in reaching that goal. It has been my observation that bad managers throw obstacles in the way of the employee, often without realizing it, and then can't understand why they don't reach the goal. Of course, they then have to blame the employee for not reaching the goal.

I have defined these roles with very clearly defined lines, but in reality, the lines blur a great deal. Managers often have to build their own teams, for example, and employees often have to perform some types of management tasks. In most small companies, a single person wears all three hats, or at least two. In virtually every company, managers must have some leadership ability to be truly effective at their job. Put another way, managers **with** leadership abilities tend to do better than those without leadership talents.

The Secrets

So finally, we talk about what the title of this paper promised.

Secret #1

You Already Know All the Secrets

This might come as a shock, but all of these "secrets" are just old-fashioned common sense, assuming there is an understanding of human nature on the part of the manager. I believe that all people, at some level, understand how to treat other people fairly. Fundamentally, employees want to be treated like people with dignity and respect, and many managers do not do this. Sometimes they don't care what their employees think or feel, and these managers are ultimately doomed. Often these people are described as "ruthless" or a "hard hitter" but in reality, they are just heartless people who only care for themselves. It is my

personal belief that the karma of these people will get them eventually. These people build a house of cards based upon deceit, fear and intimidation. Employees do not support these people, and they fall.

The other type of person who doesn't really know this secret are the kind who have never paid attention to others. Often these people want to do right by their people, but they don't really know how. Or they say things that people take the wrong way without realizing it. These people can generally be trained and develop interpersonal skills that will help them be better managers.

The key here is that a good manager must **care** about their people, and recognize that they are human beings first and employees and technicians second.

Most of "good management" is really the application of the golden rule. That is, treat your people like gold. Durable, but precious and expensive. Put another way, treat them like you'd like to be treated. I know this sounds trite, and it certainly goes counter to what most people feel is good management. But I have seen it time and again. Just put yourself in the employees' spot all the time and then act toward them like you would want to be acted toward.

Some questions you can ask yourself to help out are:

> If I were given this assignment, what information would I need to execute it well?
>
> If I'm not sure how long it would take for me to do this project, does the employee know either? Maybe I should ask him or her.
>
> What help would I need from my manager to accomplish this task?

You get the idea.

Secret #2

Learn to Manage Yourself

I've often heard the adage of those who can't do, manage. Well, a bad manager can't manage themselves. How can you tell others what to do when you can't tell yourself what to do?

First of all, I'm one of the worst at this. I seem to get everything done OK, but I don't have really good habits associated with good managers, such as making lists and so on.

Managing yourself is really a matter of self-discipline, which can be hard to start with but once habits are formed, they are easy to maintain. Here are some suggestions:

Set realizable goals for yourself. Most time management books and courses suggest daily, weekly, monthly, yearly and perhaps 5 year goals. I say do whatever works for you. Certainly it helps to set some sort of goal for the day in the morning so you know what to aim for. Without goals you will not be at all productive.

Follow your goals!

Manage your time. There are many time management courses available, some from stationary stores and others, such as Personal Resource System, through dealers. All come with some sort of time management course and paradigm. Find one that suits you and use it. This is difficult for me personally. I have a tendency to want to **do**, and not stop and write things down. Typically I can keep most things in my head, but when I have a lot of things going in a single day, I will use the daily schedule forms.

Stay calm. I tend to go non-linear when pressed for time or when pressured, but I rarely get nervous over problems. All problems have solutions, and staying calm helps to keep a clear head to solve the problems.

Find a support system. Join a professional organization and seek out people with similar careers and see how they cope with their jobs.

Secret #3

Recognize That Your People's Success is Your Success

In the definition of management given above, your results are judged by your people's results. This is a critical thing to learn when making the transition from technical fields to management.

If your company has any other definition of management, explicit or implicit, then there is something drastically wrong. An implicit definition could go by some euphemism such "Corporate Management Style" or "Management Principles of XYZ". If the managers of a company try to grind down their employees and squeeze them till they break, then they, and you, are not practicing this principle.

Secret #4

Keep Your Commitments

It has been my experience that **nothing**, repeat, **nothing** will send the trust in a relationship down the toilet faster than breaking a commitment. Stephen Covey likens a relationship to an "emotional bank account". Keeping commitments, helping people and other positive things you do for people makes deposits in this bank. Negative aspects make withdrawals from the bank. The **other** person in the relationship holds the passbook to this bank. Breaking commitments, especially repeatedly, constitute "substantial withdrawal for early penalty". Once the bank account goes negative, double or triple deposits have to be made just to get back to zero, and more to have a surplus.

The key on this one is to not make commitments you cannot keep.

Secret # 5

Ask For Their Input

It never ceases to amaze me how many people think it is a weakness to not know something. I believe it can be a strength, as long as you don't represent yourself as knowing more than you do. Managers cannot know everything, and generally employees who work with something every day will know a great deal of information that you can tap into if you just ask.

This especially applies if the project affects them. If you are uncertain how to proceed, call in some of your people and sound them out. If you take their ideas and incorporate them into the project, make sure you give them credit. (Not giving them credit may not always be bad, but taking credit for one of their ideas is sure death). Take an idea and put into the project, and they automatically have ownership of this facet. From that point forward, most people would rather die than have the project fail, since their egos are now involved.

Secret #6

Develop Goals With A Visualizable Result

If there is one thing I have learned about myself over the last few years, I cannot get motivated to work on something that I cannot visualize in my mind. (I think in pictures.) You must communicate with others what you want in a way that they can get their arms around it.

This can be tricky, since people think differently. Brain dominance theory comes into play here, and it is good for every manager to have at least a passing familiarity with this theory. Essentially, it states that each hemisphere of the brain deals with different types of symbols, and that in most people one side or the other is dominant, and thus their thinking will be dominated by the type of symbols that hemisphere processes. The right side is the artistic, picture oriented side, and the left side deals in words.
Most people are right brain dominant, and therefore respond to images better than words. However, computer programmers and designers generally have the ability to switch back and forth between the two, or to translate. Writing a program is a classic example of this. The programmer must take a picture (right brain) painted by the designer and analyst and turn this into a very specific set of commands for the computer (left brain). It is because of this ability that most technical people can respond well to goals expressed as either words or pictures. The words must be specific enough, however, that pictures can be formed. Vague goals like "this system will be written by next year" is not good enough. Detail is required.

Secret # 7

Put Yourself in Their Place

I believe this is the most important "secret".

I've already touched on this in the intro. If you are the type of manager who was promoted from a technical role, this should be easy for you. If not, then you have an uphill battle.

One thing you can try, if possible, is to actually spend a day **performing** the different job functions you manage. If you supervise operators, operate the machine for a day or two. Make this a priority. You will often find that there are inefficiencies in the procedures that the operators would not be aware of because they don't have your perspective on what else is going on in the department. Sit down and write a program and see what the development cycle is like. Is the development machine dog slow? Maybe it needs to be upgraded. Your programmers are expensive. Buying hardware might be cheap, but only if you know what they are going through.

If you are not technical by nature, perhaps you can just sit with them for a while and watch them work. Obviously, you need a relationship with them so that they don't feel threatened by this. Again, ask them what they need, and you may be surprised. They may tell you. (See secret #10)

Secret #8

Don't Treat Them Like Children

This really is implicit in several of the other "secrets" but I think it deserves special mention. Just because the box with your name in it is higher up the page on the org chart than their boxes does not give you the right to feel superior to them. Remember, your production is through them. Donald Perkins, 10 years the head of Jewel Companies, put forth the idea that the org chart of an organization should be turned upside down "so the customer was at the top and the chairman at the bottom". An interesting idea, pointing out the importance of customers and the need for the executives of a company to serve the employees, not the other way around.

I believe that if you apply the other principles, this one will take care of itself.

Secret #9

Always Think About Their Success, Not Yours

This really goes along with Secret #3. But again, I thought a bit of extra discussion might go a ways.

I would like to introduce at this point a concept presented to me be Stephen Covey, author of the book *7 Habits of Truly Effective People* (Simon and Schuster) and other books on management. He talks about a way of thinking called "The Abundance Mentality". In my mind, this is another way of saying "what goes around, comes around". If you give generously, then you will get generously. Do not think of your success per se, but give to your people so that they will be successful.

That is not to say that you should be foolish, be a doormat and let one of your people sabotage you. (Hopefully, they won't feel the need to if you treat them right). I give away a lot of free information at shows and over the phone. I could be stingy and demand payment. But my philosophy is "if it doesn't interfere significantly with what I'm doing, then I'll go ahead and give away some information". If it looks like it might take a significant amount of time, then I'll say that perhaps they need to hire me. But I'll only say this after it is clear that what the person wants cannot be answered in a conversation in a convention center hallway or over the phone. People are universally understanding, and I get some business this way. But I don't think about what business I'll get from someone when I get a phone call. I just try to help people.

The point is that you shouldn't always be hoarding your information and time, especially when it comes to your people. This also requires a great deal of self confidence, and is

not something that can be developed overnight. But the benefits are there, if you can just start down the path.

Secret #10

Develop Mutual Trust

Managing someone is first and foremost a human to human relationship. I cannot imagine how anyone can effectively manage another without first developing a trusting relationship with that person.

How is this done? Well, I would say the first thing that has to happen is you have to be trustworthy. The second is that you have to trust them. In any business relationship, there exist two types of tension. One is task tension, where energy expended in the pursuit of the mutual task. The other kind of tension is relationship tension, where energy is expended in maintaining or establishing a relationship. Initially in any relationship, the relationship tension is high, and the task tension is low. As time progresses, the task tension should increase and the relationship tension decrease.

The object of every manager is to create a situation where his or her employees work at maximum efficiency. The only way for that to occur is to make sure that relationship tension is at a minimum. This can only occur if there is trust. How can an employee work at peak when he or she is constantly looking over their shoulder waiting to be stabbed in the back? It is your task to increase this trust as soon as possible.

How is this possible? Here are a few suggestions of things I have found work over the years.

Treat everyone the same, but deal with individuals differently.

If this sounds like Zen, maybe it is. What I mean here is that you should not play favorites with your employees. Apply the same set of rules and standards to all of them. Don't let one get away with something and chastise another for the same thing. This implies that you have communicated yours and the company's expectations to them clearly. On the other hand, each individual responds better to different forms of communication than others. I've already mentioned brain dominance, but there are other factors to communication that must be considered. Some people don't want to be told all. Perhaps they are happy not knowing all the details. Some people can be convinced by rational arguments, others by emotional ones. Some, when upset, can be placated just by you being

concerned. Others need promises of things to be fixed. **(If you make them, keep them!)** It varies with the individual, and this is the core of the art of management.

This leads to

Tell them what's going on

For those of you who were hatched just yesterday, I can tell you that mushroom management is alive and well in management today. For those who, by some freak of nature, do not know what "mushroom management" is, the following laundered definitions is offered: *The act of managing by keeping the employees in the dark and feeding them excrement, a process similar to growing a mushroom.*

Tell your employees as much as you can. In fact, tell them more than they need to really do the job. Treat them like adults (see above.) It is my philosophy to tell people as much as I can think of and let **them** filter the data they need. It is much easier to hear something and ignore it than to not hear something and then have to find it out during a crisis.

Obviously, there are times when it is not prudent to tell employees **everything** that is going on. Information about projects that are still being investigated, or things that have far reaching effects on many people are often very damaging if revealed too soon. Sometimes, legal issues will prevent this, such as a pending stock split or some such thing. Don't be unreasonable. But within the realm of your authority, tell your employees everything you can. If your emotional bank account has a positive balance, then not telling people things is easier than you think. If they trust you otherwise, they will assume that the you are not telling them for good reasons, and leave it at that.

Be honest with your employees as well. If you can't tell them something, tell them that you can't tell them, instead of trying to blow something past them. Tell them as much as you can about what is going on. If you are honest with them, they will respect you. What an amazing concept.

Some Specifics for Each Type of Manager

So what now? Well, managing is an art, not a science. If I could give you a formula that would work with all people, all the time, then I'd be a wealthy man, and probably be addressing Congress, not writing papers for Interex. But I can give you some observations for the different kinds of managers.

For the techie-turned-manager, you may have it easy. If you are the rare technoid that has people skills and understands human nature, then you can be extremely successful.

If you do not possess these skills, then you must develop them or perish. Probably the most important thing you must learn, as I mentioned, is to **unlearn** that you have to **do** something to produce a result, and learn to look at what your people do as **your** result. If you find yourself programming after hours just for the heck of it, you need to seriously examine what you are doing. If you are doing OK as a manager and are just programming because it is enjoyable, you may be a bit twisted, but probably OK. If you are programming as part of your job (you are a manager which is expected to program) then that is all right too. But if you are doing this because you are frustrated and need the psychological rewards of programming, I would seriously question your career path. Ditto if you have better luck getting a computer to respond to you than another *homo sapiens.*

For the manager/manager, your main problem and/or fear is that you don't understand what your people tell you, and are worried they are blowing something past you. This is tough.

You can combat this two ways. One is to learn the profession, so that you become somewhat technical. This is good, and should probably be done anyway. But I must warn you about a couple of things. First of all, know that a little knowledge is more dangerous than none. The terror of DP departments are managers who "go to a seminar" or "go to Radio Shack" and come back full of half baked ideas. Please resist this urge. Use your knowledge to suggest and understand what your people are telling you, not to mandate new directions in the department.

Another way of dealing with this situation is to develop trust with your employees, and admit that you don't know the technical information. Establish a dialogue, and coach your senior people in how you want them to communicate with you. If all you want is the bottom line, then tell them that is what you want. If you want some detail, then communicate that. Perhaps you can find a person in the department with whom you have a good, trusting relationship and can coach them on the non-technical (business) side of things. In this way, they can communicate the issues in a way that you can understand.

Don't ignore them, though. Ride herd on them and make sure that what they are doing makes sense for the business. Sometimes techies, when left to their own devices, will spend a great deal of time doing really non-productive activities. Writing their own editor does not make sense for most businesses. Make sure you understand the business implications of what they do.

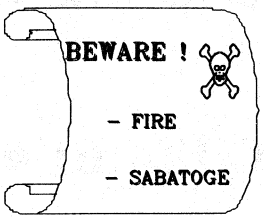
In the end, both types of managers must follow the definition of management I have put forth. Manage yourself in this endeavor; visualize the goal, and remove the obstacles to that goal.

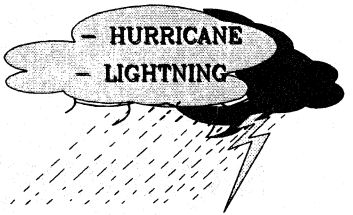
Conclusion

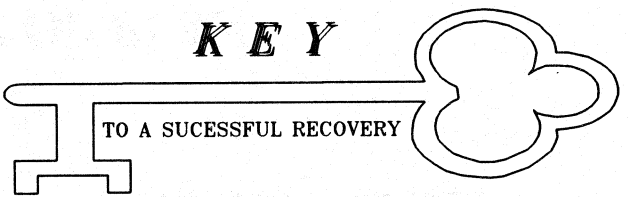
You must recognize your employees as humans, not some resource you can squeeze, grind, and run at 110% all the time. Trust me, this will only bring your own downfall. I believe that if you trust, be trusted, and follow the other suggestions here, and management will become much easier.

LOCKHEED ENGINEERING & SCIENCES CO.

J.L. HILL
2400 NASA RD 1 C30
HOUSTON, TX. 77058
713-333-7257







IS YOUR DISASTER RECOVERY PLAN A DISASTER· JH 5112 - 1

APPROACH

- DEVELOP PLAN
- SELECT SITE TYPE
- DEVELOP BACKUP/STORAGE PLAN
- RECOMMENDATION

DISASTER RECOVERY
PLANNING SERVICE

- **COMMERCIAL SERVICE**
  - **Consulting service**
  - **Experienced planners**
  - **Proven planning methodology**
  - **Recovery plan template**
- **DISASTER RECOVERY PLAN TRAINING**
  - **Experienced trainers**
  - **Proven planning methodology**
  - **Recovery plan template**
  - **Develop customized plan while you train**
- **T E S T your plan!!**

DISASTER RECOVERY
PLAN 1

- **HOT SITE PLAN**
  - **Establish hardware requirements**
  - **Determine critical applications**
  - **Develop off-site plan**
    - **Personnel**
    - **Temporary personnel**
    - **Office facilities/equipment**
  - **Test required!!**

DISASTER RECOVERY
PLAN 2

- **MOBILE SITE PLAN**
  - **Computer hardware purchase plan**
  - **Mobile-site facility plan**
  - **Utilities/contractors identified**
  - **Test not always feasible !!!**

DISASTER RECOVERY
PLAN 3

- **HOMEGROWN PLAN**
  - **Little planning experience**
  - **Anticipate most needs**
  - **Develop plan in-house**

DISASTER RECOVERY

HOT SITE

- DISASTER RECOVERY HOT-SITE SERVICE
  - Fully operational HP 3000 facility
    - Available with 24 hrs. notice
    - Telecommunications to existing users
    - Staffed with trained experts
    - Equipped office/phones/storage
  - ANNUAL SITE REHEARSAL/REVIEW
    - Fully staffed with DP personnel
    - Six days per year to rehearse
    - Vendor & MIS review/refine plan
  - DISASTER SITE RESTORATION
    - HP to evaluate equipment salvageability
    - Priority equipment delivery
    - Cold site available

DISASTER RECOVERY

MOBILE SITE

- **DISASTER RECOVERY MOBILE-SITE SYSTEM**
  - **Mobile relocatable data center**
    - **Site facility requirements/plan**
  - **Hardware/teleprocessing requirements developed**
  - **Utilities/contractors identified**
  - **Hardware independent**
  - **Rehearsal not feasible**
  - **Need open P.O. / Equipment procurement**
    - **Establish delivery date**

DISASTER RECOVERY
SYSTEM BACKUP/STORAGE PLAN

- **DIFFERENT LOCKHEED BUILDING - NO SAFE**
  - No cost
  - Not waterproof/fireproof or secure
  - Tapes too close to center
- **DIFFERENT LOCKHEED BUILDING - SAFE**
  - Waterproof/fireproof & secure
  - One time cost of safe
  - Storage continues at current site
  - Tapes subject to heat damage in fire
  - Tapes too close to center

DISASTER RECOVERY

SYSTEM BACKUP/STORAGE PLAN

- **BANK SAFETY DEPOSIT BOX**
  - Located in a back vault
  - Secure & waterproof
  - Fireproof/heatproof
  - Cannot retrieve tapes after hours
  - Some cost involved
- **COMMERCIAL DATA STORAGE COMPANY**
  - Secure & waterproof
  - Fireproof/heatproof
  - Restricted access
  - Around the clock retrieval
  - Some cost involved

DISASTER RECOVERY
COST

- **DEVELOP TRAINING PLAN**
  - **$8,000 - $10,000 TRAINING COURSE**
    WRITING A CUSTOMIZED PLAN
- **RECORD STORAGE COMPANY**
  - **$200 PER MONTH - WEEKLY DUMP PICK-UP**
  - **DAILY BACKUP TAPES TO LP4**

DISASTER RECOVERY
COST

- **DISASTER RECOVERY HOT-SITE SERVICE**
  - **$1,500 PER MONTH SUBSCRIPTION FEE**
  - **HOT-SITE (MAX 30 DAYS)**
  - **COLD-SITE (EXTENDED USE)**
- **DISASTER RECOVERY MOBILE SITE SERVICE**
  - **$1,500 ONE TIME ACTIVATION FEE**
  - **$ 295 MONTHLY SUBSCRIPTION FEE**
  - **$ 285 ANNUAL SUBSCRIPTION FEE**
  - **DISASTER FEE'S**
    - **MOBILIZATION EXPENSES ($5,000 - $25,000)**
    - **$ 200 PER DAY (MIN 30/MAX 90 DAYS)**

DISASTER RECOVERY

RECOMMENDATION

- **DISASTER RECOVERY PLANNING/TRAINING SERVICE**
- **DISASTER RECOVERY HOT-SITE SERVICE**
- **COMMERICAL TAPE STORAGE SERVICE**

THE END!

#5114

CASE in the HP 3000 Environment
Tips for Managing Your Software Development and Maintenance Life Cycle

Betsy Leight
Operations Control Systems
560 San Antonio Road
Palo Alto, California 94306
(415) 493-4122

In the last several years, many mainframe and PC software products have been grouped in the category called "computer-aided software engineering" (CASE). In some cases, these products automate program generation, others computerize the creation of specifications for programs and systems, some produce graphic representations of design, and all claim to offer the final cure for software development and maintenance backlogs. The truth however is that many companies which have purchased complicated and comprehensive systems have relegated them to "shelfware."

Fortunately for the HP 3000 user, many of the mistakes associated with implementing CASE systems have already been made by large mainframe data centers. This paper will review the lessons learned from these sites and assist the HP 3000 data center manager in handling the increasing volume of software changes required by their users.

WHAT IS CASE AND WHY USE IT?

The basic concept underlying all CASE systems is that of improved programmer/analyst productivity so that:

* Corporate system needs will be satisfied more rapidly
* The backlog of of pending applications will be reduced
* The quality of the code will be improved (both from the aspects of correctness and efficiency)
* The cost of development and subsequent maintenance efforts will be substantially reduced

The software products which fall under the CASE umbrella can be categorized into the following three classes.

Front-end CASE tools which address the systems planning, requirements definition, analysis and design phases of the systems development lifecycle.

Back-end CASE tools which apply to the coding, testing, implementation and maintenance phases of the system development lifecycle.

Integrated CASE tools which address all facets of the system development cycle.

Problems With Case

Successful implementation of full blown CASE systems often requires far more resources than most data centers have available. For effective utilization, they require a great deal of planning and training as well as integration with other tools and methods. The CASE tools have to be used consistently for both development and maintenance of the ensuing system. Procedures are a crucial part of effective system development and maintenance. Poor procedures such as not dealing effectively with client groups, improper training, inefficient standards and poor organization can offset the productivity gains that are possible with CASE.

In most situations, a CASE tool requires that standards are followed strictly, meaning that users most often return to the very front end in making program modifications to keep documentation current. Furthermore a large investment must be made in setting up a special group to design, maintain and produce periodic releases of reusable code.

Finally, much of the CASE tools available today only computerize one or more of the steps of the traditional development life-cycle. While there is some advantage to drawing a flowchart, logic diagram or data diagram using a computer rather than a pencil; these activities do not make an enormous difference in IS's responsiveness to end-user needs or timely and accurate completion of projects. As one IS manager who tried several CASE products stated, Pictures are for kids; I'll just hire programmers who can read!"

Rather than look for complete CASE systems, HP 3000 users can achieve a substantial fraction of the benefits, without the implementation hassle by focusing solely on the area of change control. Good change control can make a major contribution to the goals of minimizing bugs and other production errors, and also maximizing the productivity of MIS staff.

CHANGE CONTROL ERRORS

In our experience working with numerous MIS departments, over 50% of production problems are caused not by logic errors, but by various types of version control mistakes. The effects of these errors range from annoyance to catastrophe; nearly all could have been avoided by effective change control. Here are a few examples of version mistakes:

Fixing the Wrong Source

A program fails, leaving the sales department unable to enter orders. You are assigned to fix the problem, and, fortunately, it's easy to see what went wrong. You make a copy of the production source, fix the bug, test quickly, and move the new object into production. Users are elated. Two hours later your manager walks into your office. Your fix works fine, but the enhancement you implemented last month is gone! Apparently, you forgot to move the source back last month along with the new object when you implemented that change.

Fixing the Wrong Source, Part II

This time, you are assigned to make an enhancement to the inventory valuation system. Your shop allows unrestricted access to source code; programmers customarily make a backup copy of any source program before changing it. You make the backup copy, and begin work. You are called into an emergency meeting and asked to drop everything and work on a change to the general ledger interface which has to be completed before year-end, which is next week. Working under pressure, you are engrossed in the GL change, and forget about your half-made changes to the inventory programs. Meanwhile, an inventory valuation bug pops up. Another programmer is assigned to fix the program you had started to change. He backs up the source, fixes the bug, compiles, tests, and moves your partially completed, untested changes into production. Before the problem can be detected, numerous database records are updated incorrectly.

Concurrent Update

A programmer, John, is assigned to make a major change to a group of programs. He makes copies of all of them in his work group and begins work. Ten days later, a bug is discovered in one of the programs. A second programmer, Jean, is assigned to fix the bug. Jean, not realizing that John is already working on the program, makes her own copy, fixes the bug, tests the fix, and moves the program into production. John completes his major project several weeks later. He tests thoroughly, then moves the programs into production after coordinating database changes and production schedules. His enhancement works beautifully, but the bug Jean fixed is back.

Musical Versions

A software change requires changes to the database and to three programs that use it. You carefully build a test database and make copies of the three source files. You change the schema, make the structure changes to the test database, and modify the programs. You test thoroughly. Meanwhile your manager asks you to develop a report program using the new database structure. Finally, you move all four programs into production, and make the structure changes to the production database overnight. The next morning, it is apparent that something was forgotten: two other programs in the online system fail immediately. Major changes will be required to accommodate the new structure. Your only recourse is to roll back to the previous versions of the programs and restore the database from last night's backup. Fortunately, no changes have been made since the backup, other than your conversion. You restore the programs you changed from their backup copies - but one is missing. There is no AP1750O.PROD on disk. Someone forgot to rename the program before moving the new version into production, or renamed it to something you can't find...

These problems resulted from understandable human errors. But these errors could have been avoided by good change control procedures.

These are all examples of simple version control errors. More complex situations exist, such as the requirement to maintain multiple versions concurrently, or to maintain local changes to third-party software, and we will discuss some of these challenges later in this paper. But the majority of version control errors are of this simple variety, and can be avoided by basic change control procedure.

BASIC CHANGE CONTROL

Basic change control begins with some fundamental rules. The first of these is:

RULE #1:Production source should be kept separate from test source, and programmers should not have the access to change it except through a controlled process.

As basic as this idea is, a substantial number of MIS departments, some of them quite large, do not follow it. Many others provide for the separation, but don't enforce it by restricting access (e.g., programmers have AM capability in the account where the production source is kept). This simple control will prevent Fixing the Wrong Source, Part II", because the original source is never overwritten. The second programmer would make his own copy of the source from the original (of course, that could lead to concurrent update - see below).

The most common approach for keeping production and test files separate is called a checkout- checkin" procedure. In the checkout" phase, the programmer makes a copy of the production source in a test location, usually a separate account. After making changes and testing, the changed files are moved back into the production location, replacing the original source. The procedure is enforced by restricting programmers to read-only access to the production library. Someone other than the programmers typically performs the checkin", or move-to-production step.

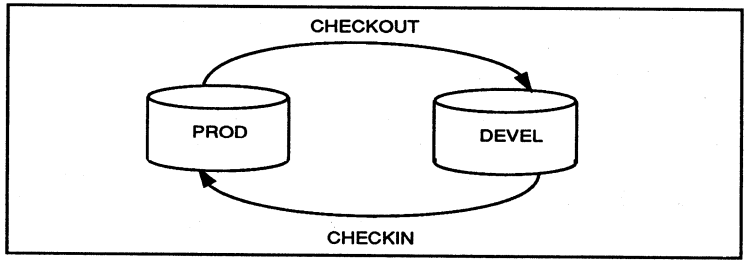


Figure 1. Checkout-Checkin Process

RULE #2:Establish a mechanism to prevent concurrent update by two or more programmers.

In many small shops, concurrent update is unlikely simply because each application is assigned to a single programmer. However, in any environment where there is a potential for two people to be working on the same program, it is a very real problem, and the bugs it can produce are some of the most insidious.

The most obvious method for preventing concurrent update is to establish some type of record of files checked out which is reviewed prior to each checkout operation. While this can be done manually, it is labor-intensive and error-prone, and therefore a candidate for automation. We have seen shops establish a physical documentation checkout in parallel with the files - the programmer signs out a program documentation binder before moving any files. If the binder isn't on the shelf, the program must be already out.

A third, more automated approach is to rename the files immediately after checkout, so that a subsequent attempt to copy the files will fail with a non-existent file error. This approach is simple and efficient, but it has three drawbacks: the person doing the check-out has to have write access to the production file to rename it, there's no way to find out who has the file if it's checked out, and the file is not found if you try to access it for any other reason (particularly a problem for mass moves). Finally, many shops prevent concurrent update by simply specifying the location in which development copies will reside and preventing overwrite through access restrictions or lockwords. If you go to check out a file and there's already a copy in the development location, you know it's already in use.

RULE #3:Keep your development area free of old copies of programs.

There are a number of ways to accomplish this, but basic diligence in limiting the source files in your development group to those being worked on is a good start. This also contributes to Part II", above, and also could have prevented the failure to update the production source in the first example of Fixing the Wrong Source". If it is exceptional for source program copies to reside in development groups, a neglected source program will stick out like a sore thumb whenever you do a LISTF.

RULE #4:Always make a backup copy of old software files before moving the new files to production. Use job streams or a software utility to reduce the opportunity for human error in this process.

A skeleton" job stream can be used effectively to make sure that the old files are renamed or copied to another group, and that the files are all moved into production. Various techniques can be used to set a flag" if any part of the move is unsuccessful. The job stream can also be set up to purge the files from the test area after a successful copy, which will help you to accomplish Rule #2, above. It might be worthwhile to write a simple utility to rename, copy, and purge a list of files. Commercial software is also available to do this.

RULE #5:Do not rely on memory to identify programs affected by changes to database structure or common code. Use a scan utility or maintain a where-used" cross-reference.

Several utilities are commercially available for searching groups of files for a particular character string such as a dataset, item, or called routine name. These utilities are easy to use, and take the guesswork out of identifying the programs affected by a change to common code. As an alternative, you can maintain a manual or automated cross-reference of programs and common code used. This will be discussed in greater detail later in the paper.

ADVANCED CHANGE CONTROL TECHNIQUES

Concurrent Development and Maintenance

While most shops will never have the challenge of maintaining multiple concurrent versions as software vendors do, MIS departments are routinely called upon to make major, long term enhancements to applications while still maintaining the current, production system. Invariably, this raises the sticky issue of how to keep the development versions

separate from the maintenance ones, and how to make sure the bug fixes and other maintenance changes made to the production system are integrated into the new version.

The first step in dealing with concurrent maintenance and development is to provide completely separate work areas for the two activities. While it may be tempting to avoid the overhead and inconvenience of working in separate accounts or groups, it is too easy to make mistakes when working with two or more files of the same or similar names in the same work area.

There are two workable approaches to forward integration of maintenance changes. One is to integrate each maintenance change into the development version immediately as soon as it is completed. This requires diligence on the part of the maintenance programmers, perhaps an additional check during the move-to-production process, and a good task list for the developers. The other approach is to make the copies for the development project at one time, take a snapshot" backup of the source library at that time, then keep track of which programs have had maintenance changes since that time. Later in the project, use a maintenance log to identify the changes that have to be integrated, or use a source comparison utility to identify the particular code changes that have to be integrated.

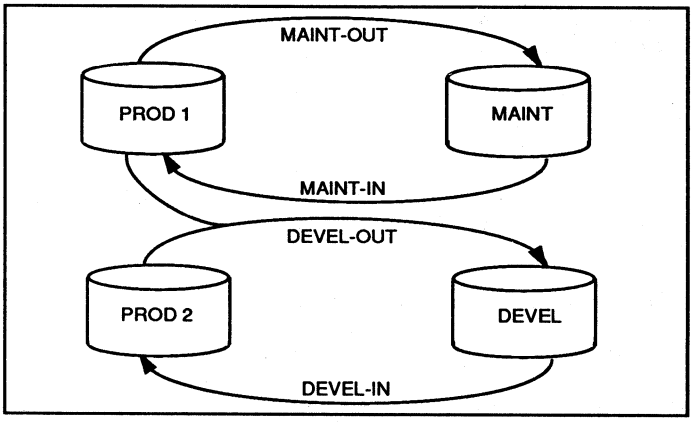


Figure 2. Concurrent Maintenance and Development

Local Modifications to Package Software

Another common, but complex, situation arises when a business makes local modifications to a software package, then has to integrate those changes each time the vendor sends out a new release. Here again, the key to managing the situation lies in keeping the modified source separate from the original vendor source. The best technique for doing this is to use the MPE accounting structure by setting up a separate group for the modified source. If the vendor's software account is divided into groups by file type, it may be simpler to set up an entire new account for the modified files, with corresponding groups. Once this structure is set up, a procedure must be established to check out files for modification from the custom group or account if they exist there, or from the vendor original

source if a customized version does not exist. When the changes are complete and tested, all source files are moved to the custom library. Object, of course, is moved to whatever location contains the executable code for the application. Most shops do not find it necessary to retain a copy of the vendor's unmodified object code.

The true test of this procedure comes when you receive a new release from the vendor, and are faced with the task of integrating your modifications into the new source. With the modified source in a separate group or account, identifying the programs that must be changed is easy. Determining what modifications must be made is not as simple. One efficient way to do this is to use a source comparison utility. Several are commercially available. The comparison utility can be used to isolate your changes to the old source. These changes can then be applied manually, or in some cases automatically, to the new source just received from the vendor. The modified code can then be compiled, tested, and moved to the modified source library.

Without a comparison utility, the best way to accomplish merging of your changes with the new source is to adhere strictly to documentation standards while making any changes to vendor code. Deleted lines should be commented out rather than being removed from the program. Changed and added lines should be clearly documented with a recognizable indicator in a particular column. If all changes are marked with a consistent indicator in the comments, a scan utility can be used to quickly list the changed lines, reducing effort and the risk of overlooking changes.

Source-object Synchronization

Source-object synchronization means ensuring that your production object code was in fact generated by compiling your production source. Source-object synchronization, combined with a good checkout-checkin procedure will prevent unpleasant surprises such as those described in Fixing the Wrong Source" at the beginning of this article.

The most common method of ensuring source-object synchronization is by recompiling the source in a controlled location. Frequently this redundant compile is performed in the production location, after checking in the changed source. In many cases the changed object is not checked in at all; the old object is replaced in the execution location by the recompile. The major drawback of recompiling in the production library is that you run the risk of putting untested object code into production. Testing is done in the development location on object code that was compiled there; this carefully tested object is then overwritten with potentially different code by the redundant compile.

A preferred method is to perform the redundant compile in a secured test area, prior to final testing. It is important that the area be secured, so that additional changes cannot be made to source after the compile, putting it out of synch with respect to the object. With this approach, it is critical that the source and object be moved together into production. This can be done procedurally, or by software that moves both files with a single operation, and moves neither if one cannot be moved for some reason.

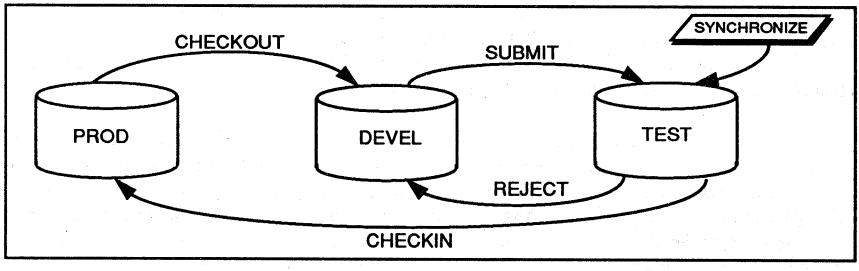


Figure 3. Secure Test Area with Source-Object Synchronization

Another approach is to mark files with a timestamp or version number to indicate synchronization. With this approach, the source and object can be compared to determine whether or not they are synchronized, rather than recompiling to force synchronization. The problem with this approach is that it is not 100% reliable unless the version number or timestamp is placed on the file by the compiling process, and stays with the file forever. A version number can be placed in the source and compiled into the object, but changing the version number is usually a manual effort, and can be forgotten. Timestamps can be synchronized by compiling, then moving the files together into the secured test area. Then timestamp synchronization can be checked again before moving the tested source and object together into production.

Another alternative is to use a MAKE utility. MAKE utilities are common in the UNIX environment, and are rapidly becoming popular on other platforms. MAKE utilities use file modification timestamps to identify object that is out of date with respect to its source - i.e., the source has a later modification timestamp than the object, indicating that the object must be rebuilt to preserve synchronization. MAKE utilities provide for definition of a hierarchy of objects", so that you can synchronize not only with respect to source, but with respect to RL's, USL's, and source INCLUDE modules in complex applications. MAKE utilities also allow you to define rebuild rules to reconstruct the object from its components in an efficient manner, only rebuilding the intermediate components that are out of date with respect to their dependencies. Since MAKE will identify objects that are out of synch, it is useful to run it against the production library on a regular basis. To avoid the trap of putting untested object code into production this way, the rebuild rules should be set up to compile into a Q/A location for testing rather than directly into the production object location. One final note on MAKE facilities: MAKE will only identify situations where the object timestamp is earlier than that of the source. If you move your object into production and leave the source behind (or purge it accidentally), MAKE will not detect this condition.

Managing Common Code

Version control for source programs and their related object files is relatively straightforward if you follow a structured, well-controlled change procedure. Managing versions of copylibs, source INCLUDE files, USL's, RL's, SL's and XL's, however, is much more complicated. If a copylib or INCLUDE file is changed, what source programs have to be

modified, or at least recompiled? If a USL is changed, what object code must be re-built? If an SL or XL routine changes, what programs may have been changed?

The simplest approach for identifying programs affected by a change to a copylib, INCLUDE file, or callable routine, is to use a scan utility. Several are available commercially (and there is at least one in the contributed library). A scan utility is used to search through a group of files for occurrences of a particular string, which might be a copybook or INCLUDE name, or the name of a called external (SL or XL) routine. The more sophisticated scan utilities can search for several strings in one pass, and use wildcard or metacharacter" search strings.

There are also commercial software packages which maintain a cross-reference index of dataset and data item names, copylibs, INCLUDES, called routines etc., specifically for this purpose. Alternatively, you can maintain your own simple cross-reference database, perhaps on a PC. Any of these approaches can prevent unpleasant surprises which result from neglecting to change all the programs affected by a database or common routine change.

To identify object code which must be rebuilt due to source INCLUDE, USL or RL changes, a MAKE facility as described earlier in this article is ideal. A MAKE facility can also do the segmentation for you to rebuild the object, based on your generic rules or rules specific to this program. A low tech", but effective alternative is to maintain a rebuild jobstream for each executable file. These job streams can be scanned for the name of the RL or USL that has been changed; those that match can then be streamed.

In addition to these questions, there are mechanics" problems, such as how to compile programs that use copylibs and INCLUDES in a test area. Should you check out the copylib and INCLUDE files along with the source? If one INCLUDE file is changed, how do you ensure that you will compile with this changed INCLUDE file but use the production version of all others?
First, many companies have found that INCLUDE files offer much greater flexibility than copylibs in managing common source code. Since each common source module is a separate file, the specific module can be checked out to make a change without tying up the entire set of common code. Moving a single INCLUDE is also much faster than moving a several-thousand-line KSAM copylib.

It is most efficient to check out only INCLUDE files that are being changed, so compiling source in a test area by pulling in INCLUDES from the production library is desirable. Read access to the production library from the development area is therefore required. Rather than having to code file equations for every INCLUDE that is not being changed, you may wish to fully qualify INCLUDE files in the source so that they point to the production library by default. That way they only have to be equated only for those that are being changed. This approach introduces some inflexibility, but a scan-and-replace utility can make short work of any mass changes that may be required.

CONCLUSION

Basic change control procedures can prevent a significant percentage of production failures, and should be part of every MIS department's operation. Change control does not have to be elaborate or cumbersome to be effective. Generally, larger shops require more sophisticated change control, but other factors are just as important. Shops with special requirements, such as maintaining local customization of supported vendor software, can benefit from more sophisticated change control techniques. Automation can improve the efficiency and reliability of change controls; you can take advantage of commercial software packages or develop your own utilities.

Paper #5115

"Don't Be Cruel to a Heart That's True"

Diane Amos, C.P.C.
Amos & Associates, Inc.
633-B Chapel Hill Road
Burlington, North Carolina 27215
United States of America
(919)-222-0231

As a manager, do you have employees whose "hearts are true"? How do you show them that you appreciate them? What do you do to retain those valuable employees?

In today's HEWLETT PACKARD marketplace, one of the costliest factors that a company has to deal with is staff turnover. The dollars start adding up when you look at the cost of recruiting a new employee, training them and then assessing lost productivity. The days of remaining in a job for twenty years to receive "the gold watch" are no longer reality, and with our mobile society being what it is, the 90's may show this to be an even larger problem. The average tenure of DP professionals is now 2 1/2 years. How can you, the manager combat this epidemic? When your good employee is resigning, should you offer him a "counter offer" to stay? NO! This is usually a drastic mistake for everyone concerned. By putting yourself in this position you are giving your employee the message that he's only valuable when threatening to leave. Let's take a look instead at preventive measures that will keep that employee from leaving in the first place. The best way to do this is to examine the numerous reasons that employees give for leaving companies.

The #1 reason for leaving a job is boredom and lack of challenge. Do you have your prize employee stuck in a corner because he does his job so well that you don't want to rock the boat? Well, the boat may sink with that attitude. People need to grow and expand in their jobs to be truly happy, and keeping a valuable employee in a "rut" will surely cause him to look elsewhere for challenges. It takes some insight and planning to keep your employees challenged, but the investment is well worth the time and effort. Think of special projects that you can assign, new and innovative ways to improve your shop. Cross training of jobs and rotation of duties keeps things interesting. In addition, many successful managers tell me that the more they challenge their employees, the more they are challenged themselves by their employees. Are you afraid to allow your employees to stretch themselves thinking that they'll outpace you? What may really happen is that you'll end up stretching yourself!

The #2 reason is lack of appreciation. When was the last time you really let your employee know how much his work, his effort and his attitude was appreciated? Do you wait for the yearly review or do you constantly reaffirm him with large doses of sincere praise when he's handed you a job well done. Management studies have shown that daily support, encouragement and praise for even the little things keeps an employee on an even keel. It's often the little things that count, and there are several small things that you can do for employees that may make a difference. Some ideas may be to take your employee out to lunch, or give him the day off. If your city has a baseball team, you might give him baseball tickets in his next paycheck. It's not the dollar amount but the thought that counts here.

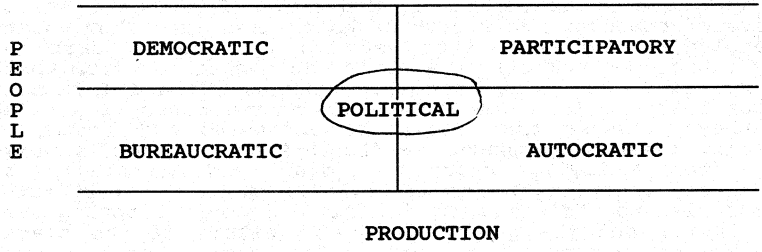
Lack of recognition is a very similar reason. When an employee goes the extra mile to finish a difficult project or solves a problem that no one else has been able to solve, he looks for some recognition that his effort was noticed and appreciated. Again, you do not have to go to great expense to let an employee know he is valued in the company, as little things that are said or done often mean more than you think. A special parking spot for the "employee of the month" costs you nothing, but makes that employee stand out. Also, choosing the employee who is doing the steady, productive job for you to go to that next LUG, RUG or INTEREX meeting rather than the same ones who are usually chosen, may give that employee the feeling of importance that is needed. Recognizing an employee in a staff meeting of his peers with a plaque for "special service" costs next to nothing. A word of caution here: it is often common practice to "take turns" in recognizing employees so that everyone gets covered. This is a mistake! It diminishes the impact and the value of the reward. Recognition is meaningless when it's given for mediocre performance and when it becomes commonplace. A very productive way to recognize an exceptional employee is to let them prepare a presentation for a user group meeting. Start with the presentation within the department. This action will recognize the employee and also train the other staff members thereby killing two birds with one stone. A success story that I can relate is one of a rather quiet, introverted System Manager who once he warmed up to you was a brilliant, funny and witty fellow. His manager put him in the position of heading up a panel at a User Group. What a pleasure it was to see this employee blossom under this new position and swell with a feeling of importance. Being creative with ways to "recognize" your valuable employees now may save you the need of looking for creative ways to replace them later.

Another reason is a lack of input in the decision making process. If Joe has a job to do but no input on the best way to do it, his frustration level will grow. People want to participate and have a voice in the decisions that affect them. Of course, that isn't always possible, but spreading the decision making power to the lower levels will instill a sense of ownership, responsibility and accountability. As Nancy Austin, co-author of "A Passion for Excellence"

discussed at the Interex Management Symposium: the companies that will grow and thrive in the 90's are those that will share the decision making power with their employees. "Trust" is a key issue here. Trust your employee to handle privileged information. You can't ask people to solve big problems without letting them solve little ones. Ms. Austin indicated that the companies who will have the **edge** will be those who nurture individual contribution, creativity and quick decision making. Critique your own management philosophy. Are you doing this?

Lack of direction from management is often heard as a reason to leave. While you're looking at your management style, check to see if you are being clear with your expectations and consistent in your demands. Monitoring your employee's progress on a monthly basis rather than at yearly review time will help to keep this in check and avoid room for misunderstandings. Have a two-way discussion to determine that you both are on the same wave length in how and when you intend to achieve your goals. Communication is the key, and having predetermined goals and objectives that you both agree on will avoid this problem. A performance plan should be in place and it's extremely important that the performance plan is a formally written document detailing the performance standards that the employee is expected to meet in order to achieve his goal. This then becomes the basis for performance appraisals which you can measure against. Employees love them, because if their boss dislikes them, they have something to put their hands around. It forces the boss to treat all his employees equitably and measure them by their performance rather than by emotion.

Matching your employees working style to your management style may help to avoid problems in the future. First of all, determine your management style. Managers should be both people oriented and production oriented. Look at the diagram:



Is your style Democratic where you have more concern for your people than for production? Is it Autocratic where you care more for production than for your people? Or is it Participatory where you're concerned for both? Or Bureaucratic where you don't care about either your people or production? Or is it Political where you flip-flop between all four styles at any given time? Having your applicant describe his ideal boss may help you to make that right hire and match the style at the very beginning. This may save you endless struggles of philosophical differences that no one can solve.

No career path? I hear this reason for leaving quite often! This is a difficult area in the HP market because there are so many small shops with few vertical career paths to offer. People tend to think of career paths as straight up when sometimes they're sideways and at angles. Discuss openly and freely with your employee what he truly wishes to do with his life and career. Perhaps you'll find that he could progress into the user area and would be happy to do so. Then put a career plan in place where he can take concrete steps to prepare for this next step. Don't make empty promises, but give a clear direction on how he can reach that goal. Make it a formally written plan detailing what your employee wants to do, and what he must learn or achieve to get there. Does he need to take a certain course or learn a particular skill? Where will he be able to go if he masters that? Be very specific and clear. What if your valuable employee wants to progress, but stay in a technical field and not go into management? A long term employee brings added value to the company and he shouldn't have to change jobs to have a career path. Raising the upper limits on salary and increasing the responsibilities and decision making input for people in the technical field could keep these employees satisfied.

A poor working environment is often given as a reason to leave. As manager, take a hard, critical look at the workplace that you have to offer. The best way to do this is to actually work in it for awhile. Is it conducive to good work, and a place where people will enjoy coming to work, or does it leave something to be desired? I've heard horror stories of computer rooms with desks crammed into them making the employees feel like they were in jail cells; dark and dingy cubicles stuck off in a corner somewhere far from where they needed to be to be effective with users; and I've seen offices that should house 1 programmer have 4 people stuffed into them. Issues that should be addressed and looked at carefully are the temperature - (too hot or too cold?), noise level- (too noisy?), color - (drab and lifeless, or offensively unnerving?), smoke-filled?, music - (irritating or conducive to efficiency?), traffic - (people tripping over each other?) and the general esthetic nature of the place. These are often more subtle reasons for leaving but they work on the frustration level of an employee. Have you ever been bitten by an elephant? No, it's the gnats that get you, not the elephants.

Are you giving your employees valuable training and sending them to the necessary schools, or are you expecting them to pick up everything on their own? It's been said that people are most happy doing what they know how to do well. If they feel like it's a constant struggle to keep up because of lack of training, this frustration may turn into a reason to leave. Giving training is a good investment as it allows your employee to be more productive quicker than if he had to learn something from scratch. An important side benefit to his productivity is his feeling of putting to use what he learned and satisfying his thirst for knowledge. An inexpensive and productive way to do this is to provide in-house training by one of your own "experts". This idea cuts costs and recognizes a star employee all at the same time.

With all of the latest technology bombarding the HP market, a common complaint by employees is not being trained on the "latest, greatest". Employees want to stay current in their knowledge of technology and resent working on out-of-date equipment or unsatisfactory software. They know it's career suicide to do otherwise. Employees will jump ship just for the opportunity to work in a "state of the art" shop. As managers, staying current yourself with what's new and desirable on the technology front will keep you ahead of the game, make your shop an attractive place to come to and one that people will not want to leave.

"I'm stuck with a boss that hates me". What do you do when you have a personality conflict? IBM Corporation gives "skip-level interviews". This is where the boss's boss meets with the employees to hear what's going on. One benefit of this is for the senior manager to "hear from the horse's mouth" what's going on in the department, rather than only what the department manager wants to report. A critical benefit for the employee is that he has "an ear" for his complaints and an outlet for his frustration. Second-level management reviews of performance appraisals work well too. This is where the boss's boss sees the appraisal before the employee does, and then again after the employee is given the appraisal, but before it goes to Personnel. This offers good objective safeguards.

What if you have a "bad apple" who's driving everyone away? We've all seen this happen, and it always seems to take numerous sacrifices of excellent employees before management wakes up and does something about the "bad apple". Bad employees add to the work load of the good employees, and the good ones end up being "punished" for someone else's failures. This is sad to see, and it's often symptomatic of the manager's inability to deal with problems. As a manager, you can prevent this from happening by being decisive and cutting the rope when necessary. Keeping someone on board who spoils the rest of your staff is only going to cause you more problems in the long run. No one is indispensable and even if this "bad apple" is the most knowledgeable person you have on staff, cut your losses and do it quick! Remember, it's not the people you fire that give you trouble, it's the people that you don't fire.

Is your shop an unhappy place to work? Make it fun! Work does not have to be serious, and laughter cures all ills. I've seen more contented employees in companies just because it was a fun place to be, and everyone had a good time working there. Be creative! Start your next meeting with a video of "baseball bloopers" to get everyone in a good mood. And who says that a department meeting has to be in a conference room? Take everyone out to the park and have them bring their lunch. If you're in a coat and tie environment, have "casual day" every other Friday. On your way to work, stop by the 7-11 and pick up the big 50 cent bubble gum. That will turn everyone into kids when each person compares how big their bubble is. Dress up for Halloween. Have a team golf or volleyball game. An owner of a company once said to me, "I feel that I'm in the entertainment business-making my people happy." As manager of your department, you can make this happen too.

$$$$...Salary is a common reason given for leaving although I believe that it is generally secondary to the other reasons. Sometimes you just have to dig deeper to uncover the real reasons. Too often they only come to light at the exit interview when it's too late to remedy the situation. If someone is unhappy with his job, he may say to himself.."I'm not paid enough to take this abuse". But I've seen too many times where an employee will stay with a company for a long time, underpaid, simply because he is truly happy with his working conditions, treatment and job functions. However, let's address the salary issue. Is pay based on performance more than on seniority? It better be or you'll end up keeping your oldest employees instead of your best ones. Have you been staying current with the going salaries in the marketplace or have you fallen behind? Has the employee's job grown to where it needs to be reevaluated and a new worth established? Assess each job objectively and decide what it is worth. Then pay accordingly! An added possibility is the incentive bonus. I have a client who uses the incentive bonuses very successfully and reports a dramatic increase in productivity. He and the employee set concrete and clear goals at the beginning of the year, with several goals (8-10) to reach and then monitors the progress monthly. He says the employee has a sense of pride, ownership and reward when the goal is achieved, and the money turns out to be only a side benefit.

Let me leave you with a story. There was this data processing company that was seeking a technological breakthrough which would put their company out in front in their industry. Because this was a make or break situation for this company, the entire staff was hard at work to achieve this goal. One day the President of the company was in the lunchroom having lunch, when one of his technological genius's suddenly found the answer to the problem. He had broken through the barrier and discovered what they were all looking for! The President was obviously thrilled and

excited beyond belief, and wanted to instantly reward this employee for this achievement. In his excitement, the only thing he could think of saying was, " Here. Take my banana!" Well, the employee was so pleased with being recognized by the President, that he put the banana on his shelf for all to see. It turned brown, started to smell, but he wouldn't throw it away. When the President saw what the employee had done with the banana, the symbolism hit him and he immediately went out and bought little banana pins. He went back to that employee and pinned the little banana pin on him. From that point forward, the 1 cent "Top Banana" pin became the badge of highest honor in that company.

The moral of the story is corny but real...don't be cruel to a "heart that's true" if you want to keep your "top bananas"!

PAPER NUMBER: 5116

TITLE: EXPANDING YOUR COMPUTER OPERATIONS INTO EUROPE ISN'T NECESSARILY A VACATION!!

AUTHOR: JOHN R. BEDARD

COMPANY: ST. JUDE MEDICAL, INC.
ONE LILLEHEI PLAZA
ST. PAUL, MN 55117
612-483-2000

One thing about experience is that when you don't have very much you're apt to get a lot.

Franklin P. Jones

When our company decided to establish a direct sales force in Europe, Information Services was given the challenge to install the necessary computer system and related network to support the new European headquarters and five branch offices in less than nine months. This meant that we had to design the system and order all equipment before any specific locations were determined and before we had hired any European personnel to assist us. In fact, the European staff was being hired just as we were installing the system.

Since no one in our department had experience in setting up a computer network in Europe, we got a lot of it. (Maybe even enough to become consultants.) This paper will explain the many challenges that a company expanding its computer operations to Europe for the first time faces and how we dealt with them. They include:

- Ordering for European delivery (European customs means more than which side of the plate to put the fork on)
- Data Communications Costs and Lead Time (Dealing with governmental agencies can leave you with a failure to communicate)
- European Hiring Practices (Do you want resumes in Dutch or French)
- Ordering the correct equipment (Do you want Qwerty or Azerty keyboards)
- What support can you expect from HP (I'm still not entirely sure, and I don't think they are either)
- Getting the right option (How can Europe agree on a common currency when they don't agree on a wall outlet)
- European Business Practices (Life's a holiday)
- Getting the computer installed (Some assembly required)

One thing our experience on this project taught us is that Murphy is alive and well and his law applies in every country in Europe!!

ORDERING FOR EUROPEAN DELIVERY

There are multiple ways you can order HP equipment for installation in Europe. We chose to order through our local (St. Paul) HP office at export prices. This gave us the advantage of working with our local HP support team who knows our company and our existing configuration and applications. The local office also understood the pace which this project had to proceed. I suggest that anybody who is considering a European expansion take time to make sure that they know what the ordering options are and make a decision on their particular needs.

As we proceeded with this option we received a number of surprises. In some cases we were told about these issues but did not understand their implications until we had to deal with them. These issues included:

- **Longer Quote Lead Time**

  We could not place orders for international delivery until we received quotes from HP Intercontinental Operations (hereafter affectionately referred to as "ICON"). These quotes always took at least two weeks, and usually much longer to obtain. Make sure you allow sufficient time for export quotes.

- **Increased Costs**

  In our preliminary budget estimates we used domestic pricing. When it came time to actually order the hardware and software we got the ICON quotes. They were about 25% higher than the domestic quotes we were using for budgeting purposes. In this and many other areas we found the cost of doing business in Europe was considerably more expensive than doing business in the U.S.

- **Clearing Customs**

  HP ships to your international address. However, your shipment will sit in customs until somebody arranges for clearance, including payment of all duties and VAT. (You will learn a lot about VAT in your European experience.) Our customer service staff, which exports our product to Europe helped us with this function. In some countries we also used a customs broker. Make sure that you know who will perform this function and allow enough time for clearing customs.

  In clearing customs you will be dealing with a government agency, so all paperwork has to be exactly correct. As we had multiple legal companies involved in our European operation this became complicated. The penalty for incorrect paperwork will be delay and possible loss of the ability to recapture VAT.

- **Export License Requirements**

  Since the HP 3000 computers can be a terrible weapon in the hands of an enemy, the U.S. government requires export licenses for most HP computer equipment that is exported. Plan on learning more about the legal paperwork or have one of your company's departments that know about export licenses help you. (A real payoff from the good service you have been giving that department.)

- **Volume Purchase Agreements**

  If you have a volume purchase agreement, make sure it covers your international shipments. They are not automatically included.

- **Delay in Correcting Ordering Mistakes**

  Virtually any ordering (or shipping) mistake has to be corrected by a new shipment from ICON. So it takes awhile to correct. Don't believe those ads that say you can ship next day to Europe like you can in the U.S. It will take several days just for HP to get all its internal paperwork straightened out.

- **Coordinated Deliveries**

  Make sure that everything you need is on a coordinated delivery. And don't forget anything (like terminals or PCs) that may be shipping to a country different from the country where the CPU is shipping. Keep everything on a coordinated delivery and you will get it all about the same time. We forgot to do this on our remote equipment and had to scramble (and beg and threaten) to get our remote location equipment on time.

- **Delivery Address**

  This presented an interesting challenge for us as ICON will not accept an order without a delivery address. But the lead time on our equipment forced us to place the orders before we had found locations for our offices. So we used the address of the HP office in each city that we planned to have a facility as our shipping address on each order. Fortunately our facilities people did arrange for our office space in each city before shipment and before several European HP offices received unexpected presents.

DATA COMMUNICATIONS COSTS AND LEAD TIME

We faced two main challenges in trying to establish our data communications network in Europe: cost and lead time. In most countries the phone company is not only still a monopoly, but is part of the government postal system. (How would you like to order your data communications lines from the U.S. Postal Service?) We discovered that a leased line from Brussels to Paris would cost three times what a

leased line from St. Paul to Boston cost us - and it is only about one-third the distance!!

Our data communications traffic did not justify leased lines so we decided to use the X.25 network of a Value Added Network (Tymnet, Infonet, Compuserve, etc.). This approach also let us deal with a single vendor, instead of having to deal with each country's phone company (post office). As we installed our network we continually faced new challenges:

- **Lead Times**

  We used leased lines to connect to the vendor's node sites. Lead times in many European countries is three to four months for these leased lines. Also, in many countries we could not even get definite installation dates until about a week before the phone company was ready to install. Even after a line is installed you can experience maddening delays. We were repeatedly told that a relatively minor change in our Spanish service would be done "mañana".

- **Government Restrictions**

  Spanish law forbids the value added networks from offering this type of network service. So in Spain we had to contract with the government network service and interconnect to our vendor's network — an added level of complication. (We were also cautioned against trying dial-up service in Spain because of quality problems on their voice network.)

- **Modem Restrictions**

  In Belgium the phone company does not allow a computer to place a call (as in Predictive Support). Also, none of the modems approved by the Belgium phone company are supported by HP on the Openview DTC workstation.

- **Vendor Inexperience**

  As our project proceeded we realized that we were among the first users of this specific network service of our vendor. Next time we will make sure to check out the vendor's experience with the exact service that we will be using. The vendor's learning curve cost me several gray hairs (and loud phone calls).

EUROPEAN HIRING PRACTICES

Have you ever tried to interview someone who does not speak English when English is the only language you know (not counting COBOL)? While many Europeans are multi-lingual, it is important to determine language requirements before you begin the hiring process. Besides languages, we learned there are other considerations in staffing in Europe:

- **Notice Requirements**

  It is not unusual for a person to have to give his employer a minimum of six weeks notice before he can quit his current job. In some cases it can be a year or more.

- **Staffing Levels**

  We had to use a different staffing strategy in some departments because of the number of holiday and vacation days that everybody gets.

- **Job Titles**

  In one market we found that the job title "Computer Operator" meant much less responsibility than that same title in our U.S. operations. Consequently, we changed the title to conform to local custom.

ORDERING THE CORRECT EQUIPMENT

Since English was going to be the language of the company, and all computer screens were going to be in English, it only made sense to order English keyboards. Or so it seemed to us here in America. As soon as we hired the first secretary for our Belgium office we learned that the French language keyboard is AZERTY compared to the English QWERTY. We also learned that a country manager will agree with his secretary rather than with a IS manager in America. Other lessons that we learned about ordering the correct equipment included:

- **LAN Wiring**

  We cannot assume that the phone wiring will support a twisted pair LAN like we can here. Fortunately I checked this out ahead of time and did not suffer the well-known consequences of assuming.

- **Environmental Control**

  Our facilities manager believed the leasing agent when the agent told him that the office was air conditioned. (Actually, I think our American facilities manager misunderstood him, since air conditioning is not as prevalent in Europe as in the U.S.). There was some degree of environmental control, but not enough for a computer. After HP did their site survey we scrambled to get an air conditioner installed.

- **Order Everything Complete and Correct**

  As discussed, under ORDERING FOR EUROPEAN DELIVERY above, it can take longer than you can afford to correct an ordering mistake.

WHAT SUPPORT CAN YOU EXPECT FROM HP

As I write this, I realize that I am not entirely sure and I am not sure that anyone at HP can give you a definitive answer either. After this experience, I am coming to the conclusion that HP is not really one company. (But then I sometimes get that feeling dealing with various parts of HP here is the U.S. also.)

I will say that I am grateful to HP both here in the U.S. and in Europe. Several people from HP on both sides of the Atlantic put in extraordinary effort so that we could have a successful, on-time, installation (which made me look good in the eyes of my management).

Suggestions that I would offer to someone on a similar project include:

- **Make Sure HP Agrees with HP**

  Originally, I was told that HP in Europe would be responsible for all installation. But HP in Europe wants a NISP (Network Installation and Support Plan) prepared by HP in St. Paul. Only after I complained to the local HP Field Engineering management about being in the middle of two HP organizations did appropriate HP engineers on both sides of the Atlantic begin to communicate with each other.

- **Put Plans in Writing**

  Make sure that the objectives and plans are in writing and understood by all parties. It is very easy to miscommunicate with the people with whom you work and meet with regularly; it is almost impossible not to miscommunicate when coordinating a transatlantic installation verbally.

- **Be Careful of Changes**

  For several reasons we made changes to the number and mix of terminals and PC's from the time of our initial plan until the actual installation. HP in Brussels had the wiring done to the plan we gave them rather than to what we actually ordered. We thought (assumed) that the local HP office would notify Brussels of the changes when we placed the final order. (Now I suffered the well-known consequences of assuming.)

- **Each Country Adds Lead Time**

  HP Brussels coordinated all quotes and service with other HP offices in Europe. However, from the pace at which we got information relative to the other countries one might have thought the other countries were on different continents.

GETTING THE RIGHT OPTIONS

One of the truly exciting features of ordering from HP is the interesting combinations one can put together from their list of options. Our first shipment of PC's to Brussels had 220 volt processors with 5.25" disks, 110 volt monitors, and software on 3.5" disks. The Brussels Field Engineering manager rescued us by scouring PC dealers for enough parts to let us assemble a few PC's until we could get the order corrected.

Remember that not all countries use the same power outlets, some countries have multiple languages, and there are language options on hardware (keyboards) and on software (e.g. WordPerfect).

EUROPEAN BUSINESS PRACTICES

My observation is that in general the pace of business is slower in Europe and the government requirements require greater consideration in system design. Some of the practices of which you should be aware are:

- **Vacation and Holiday**

  In some countries it seems like the whole country goes on vacation for a specific month in the summer and from December 15 to January 15. And we half-jokingly ask our European offices what holiday do they have off this week. (But they do not celebrate July 4 in England.) Holidays and vacations have to be taken into account in any project planning for Europe.

- **Accounting System**

  It is not unusual to have requirements such as not having receivables in a subsidiary ledger, but right in the general ledger (meaning every customer is a general ledger account). Just before we went live with our distribution system we discovered that we had to provide a sequential control number for invoices in each country, in addition to the system-wide sequential control that we had provided in the system. We also had to print more copies of paper documents for government auditors who have not heard of (or do not believe in) microfilm.

- **Format Issues**

  There are several format issues in Europe, some European-wide and some country specific. These include date formats, currency symbols and decimal symbols (and whether or not there are decimals in a particular currency), size of currency fields (see what it does to your database to accommodate Italian Lira), paper sizes, and window envelopes. Our German office initially retyped all of the invoices for German customers because they did not like the location of the window of our standard window envelope.

INSTALLATION

My basic advice if you are doing an installation with some of the same constraints that we had is to contract with HP for complete installation and let them subcontract as appropriate. We contracted for Netstartup and Netassure so that we would minimize the number of companies we had to deal with and to provide a clear focus of responsibility. We also contracted with HP for all terminal and LAN wiring, computer room power management, and computer room environmental control.

GENERAL CONCLUSIONS

As I reflect the experience that I gained on this project, I come to the following general conclusions:

- **Lead Time**

  In real estate the three most important factors are location, location, and location. In a European installation project I would say the three most important factors are lead time, lead time, and lead time. Remember that you cannot even order a phone line (and start that clock ticking) until all your legal organizational papers are complete. And if the data communications isn't in on time, they won't remember the amount of lead time you told them you needed to install data communications.

  If ever there was a place to apply Deadline Dan's Demon, this is it. (Every task takes twice as long as you think it will take. If you double the time it will take, it will always take four times as long.)

- **Personal Contact**

  You can't do it all on the phone. Remember, you only have a few hours a day during which your workday and Europe's workday overlap. You will get much better results working with people with whom you have developed a relationship. In retrospect, the most important thing I could have changed in how we did this project was to go to Europe earlier in the planning process and to go more frequently to monitor progress.

- **Focal Point**

  Make sure that there is one person responsible for dealing with a vendor. I was telling our Value Added Network vendor that I needed our French data communication connection in by December 1 so I could conduct the necessary testing. Meanwhile, someone in our French office told them we didn't need it until January 1 because that's when we needed to go live. (Guess which date the French phone company aimed at?)

- **Have Someone Knowledgeable On-site**

  What is apparent to you isn't always apparent to others. In our Spanish office the secretary decided that the datacomm modem and multiplexor was expensive equipment and should be safeguarded, so she told the phone company to install them in a private office — which was a great distance from the point where our terminal cables terminated.

- **Murphy's Law**

  Whatever can go wrong, will go wrong. How could I know that Desert Storm would preempt all Laserjets from HP and I would have to find dealers in Brussels who had them in stock?

- **Murphy's Law of Thermodynamics**

  Things get worse under pressure. I had four hours in Dusseldorf to check out all equipment and data communications in that office. Then I lose an hour because of a cab driver who cannot find the address. (And of course he cannot speak English and I don't speak German, so we raise our voices to try to increase understanding.)

Actually, I feel that I have gained more than just experience on this project. I gained 10 pounds from all the Belgium chocolate I ate while in Brussels installing the system.

Judgment comes from experience, experience comes from poor judgment.

Robert E. Lee's Truce

Footnote: If everybody reads this paper before coming to the session, we can devote most of the session time to a review of recommended restaurants for your trip to Europe.

UNDERSTANDING POWER
"AN INFORMATION SYSTEMS MANAGEMENT CHALLENGE"

Robert Berry, Ed.D.
Administrator, Information Systems Department
Jeri Wenger
Administrative Supervisor, Information Systems Department
San Bernardino County Superintendent of Schools
601 North E Street
San Bernardino, CA 92410
(714) 387-4466

Introduction

Implementing and managing information technological changes in the 1990's will cause fundamental structural changes within the Information Systems (IS) Department. Rather than adapting to the environmental changes which will occur, IS managers must take a proactive approach to control the change process by utilizing the power inherent in technology. Success will come to those IS managers who can proactively conceptualize the fundamental technological structural changes as they occur within the business environment and utilize knowledge of power to implement these changes. This is an Information Systems management challenge.

Traditionally many IS departments derive their power base from the control of technology. Technology, as historically developed, was centralized and controlled by the IS department. Technology provided its users with an ability to achieve a high degree of productivity and to manipulate this productive power. Therefore, many organizations have become vitally dependent on some form of core technology.

Today, however, many IS professionals do not understand that due to the technological advances that have occurred, the control over the use of technology is no longer isolated to the IS specialist. More and more the end user is a computer literate person that is demanding a bigger role in the decision-making process concerning the use of technology. This changing base of knowledge concerning technology and its uses has had a major impact on the power relations within an organization. This is an important reason why attempts to utilize technology often create major conflicts between IS and other groups within the organization.

Management models have been created in the past as "tools" to assist managers in identifying changes and implementing solutions. These older models, however, are no longer appropriate for managing information technological changes in the 1990's. The technology power base is shifting rapidly away from the IS department, and IS managers must now focus on what is required to regain this power base.

Based on research, observations, and experience, we have developed a Knowledge Processing Plan Model. Although our model is very broad and encompasses several management components, the focus of this article will be centered strictly on concepts about power, sources of quality power, and methods which, if used effectively, will allow IS managers to accrue power and use it constructively. In order to use power effectively, to think strategically, and become proactive, it is important to understand what power is and how it relates to other aspects of the business environment.

Exploring Power

To be successful in IS management today, it is important to understand that power should be considered a resource to acquire and use. Power should not carry a negative connotation, unless you choose to think about power in a machiavellian sense. Many information systems managers, and in fact managers in all facets of business, accrue power and use power constructively every day. Robert Dahl, an American political scientist, suggests that power involves an ability to get another person to do something that they would not otherwise have done. Another organizational theorist, Gareth Morgan, writes, "Power influences who gets what, when, and how." Within this framework, power should be considered a prime ingredient in the manipulation of resources that takes place in everyday business.

It is difficult to define exactly what the phenomenon that we call *power* is. Because the concept of power is, by itself, abstract and filled with both personal and political controversy, it becomes important to review a few basic assumptions that can be used to assist in the definition of the concept of power.

Power Assumptions

1. Power is inherent in all social systems and in all human relationships. It is not a thing but an aspect of any and all relationships among people. Hence, it is inescapable and neutral, intrinsically neither good nor bad.
2. Power struggles are not necessarily bad.
3. Power is a resource that can be acquired, measured, and used.
4. The quality of one's power is more important than the quantity of power. For often the quality of a power resource will in fact provide a measure of control over a superior quantity of power.
5. Mechanisms created by managers to control authority and power over subordinates in fact create power sources for subordinates to control management.

Since the premise of this article is built firmly upon these assumptions, it is important to note at this point that to understand the concepts of this article, one must accept these assumptions as truisms. If not, then there is no need to read further.

While each of these assumptions is interwoven into our definition of power, the underlying framework for this article is built primarily upon assumption 4: The quality of one's power is more important than the quantity of power. For often the quality of a power resource will, in fact, provide a measure of control over a superior quantity of power.

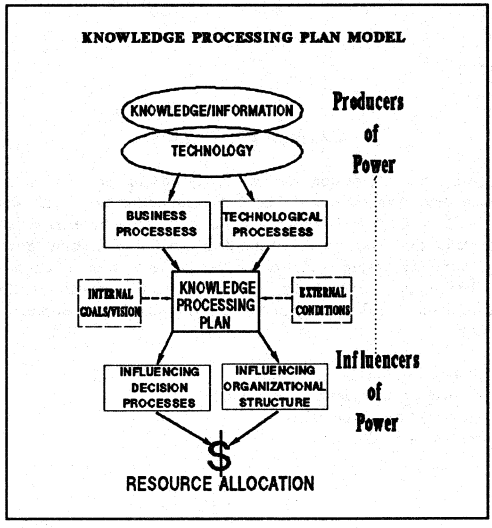
The IS professional who understands "quality of power" will have a strategic advantage in the power struggle over the control of technology. To expand on this concept, it is necessary to understand where key sources of power can be acquired and how they can be applied to achieve a desired outcome. Understanding these and the relationship among them will define power for the IS professional.

Quality Sources of Power

While organizational and management theorists have not been able to identify clear and consistent sources of power, we believe the IS professional has access to and the ability to influence the following quality sources of power:

Quality Sources of Power	
1.	Knowledge and information
2.	Technology
3.	Influencing organizational structure
4.	Influencing the decision processes

Although the accumulation of any one of these sources will provide the IS professional with moderate power, it is the combination of these four sources that will yield the highest quality of power. Quality power, once obtained, will provide the IS professional with the ability to *control resource allocation and create a dependency*....the ultimate prize in the corporate power struggle. The combination of these four sources constitutes the key elements comprising the framework for our knowledge processing plan model.



As illustrated, the four quality sources of power are separated into two distinct categories within our model: *producers of power and influencers of power.* At the top of the model resides knowledge and information intertwined with technology; *the producers of power.* At the bottom of the model resides the decision processes and organizational structure; *the influencers of power.*

Although the model appears to be hierarchal, the producers of power are the essential foundation of the model and, therefore, represent the highest quality of power. In general IS professionals have acquired knowledge and information and are very astute technologically, but have not applied these sources of power to the organization's business processes. Although we understand and have developed long-term plans, more often than not the organization's business and technological processes have not been properly aligned with the internal and external environmental conditions. Ironically, long-term plans tend to accumulate dust and are not utilized or seen as power instruments.

Traditionally, management schools have taught professionals proper methods for "producing" a plan but have fallen short of clearly defining methods on how to "utilize" the plan as an "influencer of power" to achieve a desired outcome. In some cases the technology power base is shifting rapidly away from the IS department. Therefore, it is critical that IS professionals understand how to use a "Knowledge Processing Plan" as a change agent to influence the decision processes and organizational structure to acquire the resources necessary to effectively implement the plan.

Implementation of the knowledge processing plans should help to ensure that the business remains competitive and successful in today's rapidly changing business environment. We have provided a brief outline highlighting the major components of our model: *the producers of power and the influencers of power.* To fully understand how to effectively utilize this model as a power instrument, we will explore each of the components separately and in more detail.

Producers of Power

Knowledge and Information

As stated earlier, knowledge and information are the very foundation of our model. Just as technology has changed radically in recent years, so has the user's knowledge and information level. Many IS professionals have not realized that the users they design systems for and attempt to serve have more information and are more technologically knowledgeable than ever before. By not understanding this, many IS managers continue to play the role of knowledge gatekeepers, trying to control the opening and closing of channels of information about technology and company resources, thus influencing who gets access to what data or information. In some cases IS professionals have even controlled the quality of data released to other co-workers. IS managers have been able in the past to control:

- What data or information is available
- How the data or information is structured
- Knowledge that data even exists
- Who gets access to the data

The days of the "Technowizard" are over. No longer is it necessary to totally rely on the IS department for access to information. The combination of microcomputers and networks has forever changed the basic power structure of the IS department. In a general sense, the microcomputer, when tied into a local area network and/or wide area network, allows for the control of knowledge and technology to shift to the periphery of an organization. Knowledge is power, but knowledge about knowledge yields the best quality of power. The proliferation of databases and information services allows all levels of management to access and gather information never before available to them. This new form of information gathering and processing power points to a revolution in our ability to think, analyze, synthesize, and express our knowledge processing skills.

IS professionals should now strive to take on the role of in-house consultant. We should strive to broaden our perspective and thinking skills to look down all avenues of possible solutions prior to recommending technological solutions. We must teach and allow our users to participate in information gathering and increase their ability to think, analyze, and synthesize information concerning their own business requirements and needs for technology. This implies that a new employee skill set needs to be developed within information departments.

IS managers must understand this fundamental shift in the thinking skills that are now required concerning knowledge processing and knowledge acquisition. We must provide our employees with the opportunity to acquire problem solving skills, meeting facilitation skills, leadership skills, and other managerial skills. Those IS managers who develop staffs with these skill sets will more often win the corporate power struggles.

Technology

"Our organizations are killing us . . . or are we killing ourselves?" Indeed, the first part of this statement reflects the view of many IS organizations as they attempt to continue to control technology within their companies. Unix, open systems, network standards, downsizing, and outsourcing all are the hot topics of the day. Many of these new hot topics not only reflect the ever-changing technology, but could be thought of as a battle over the control of computing resources. In fact, the classic battles over centralized computing versus decentralized computing were not only battles over technology, but were also battles over control of information and knowledge about information.

Prior to the advent of the microcomputer in the 1970's, IS professionals had centralized control over the processing power of their companies. What has transpired, however, is that the microcomputer technology has fundamentally shifted the power base away from the IS professional to the end user. Today in many firms the combined power of the microprocessors probably would account for more than 50% of the total company's processing power. Stonewalled by IS professionals who thought of microprocessors as insignificant, many users were end-running the IS departments and setting up their own isolated decentralized processing stations to solve many of their own informational needs. Along with the hardware acquisition came knowledge acquisition about technology that the IS professional controlled. As a result of this shift of power, users are no longer mystified by basic computing technology or terminology.

Today many IS professionals are trying to regain control over opportunities lost. Different computer systems, different databases, isolated knowledge processing systems all are killing the concept of an organizational computing resource plan. Many department managers have been very clever in building into their budgets adequate resources to continue to exert their computing independence. Under the guise of the electronic highway, the IS professionals are attempting to recoup some order to the anarchy created by everyone "doing their own thing." IS professionals today are embracing the microcomputers as never before, while at the same time attempting to re-assert control, hence, power over their acquisition and use by enforcing networking standards, i.e., the connectivity war.

To do this successfully, the IS professional should incorporate the technology plan as only a part of the knowledge processing plan. Technology, no matter how elegantly implemented, will not, by itself, curtail the power struggle over its control. We are killing ourselves when we use technology as a club to enforce the rules or use technology as an excuse for organizational failure. Technology is not the problem. It's our ability to control and channel its vast potential to assist in the influencing of the decision processes and organizational structure.

This attempt to recapture control will again have a major impact on power relations within the organization. The control over technology often creates major conflicts between IS and other groups within the organization. IS professionals must understand the fundamental shift away from trying to control technology to becoming enforcers of company standards. Another battle is currently underway which raises several issues for the IS professional to consider. Will the IS professional view the advent of telecommunications, i.e., video, voice, and data as insignificant to the IS department much like microcomputers? Will the AT&T "death star" also be the IS death star? Will some IS departments suffer major power losses to the new kid on the block -- the Telecommunications Specialist? Where does this resource belong? The technology base is shifting again and the IS professional must determine if this is to be another opportunity lost.

Influencers of Power

Organizational Structure

Usually organizational change revolves around the shifting of power and resources. The tensions surrounding the process of organizational design and redesign can provide the IS professional with valuable insight as to the existing power structure. Too many times a new technology is introduced without changing the organizational structure. As Venkatraman, from the Sloan School of Management, MIT, states, "The increased use/purchase of additional technology simply placed over existing conditions returns very little (if any) return on the investment."

While technology excels at improving coordination, access, and analysis of information, technology alone is not the total solution for the long-term health and success of the organization. By using technology as a corporate foundation, the IS professional can apply technology to interorganizational tasks which will require the organization to develop a corporate vision and strategy. Therefore, the role of IS should be to consult with the divisions and departments, set policy and standards, review and approve system plans and expenditures, and attempt to establish a corporate vision. This type of role will require a power shift of authority and often blur the lines of authority that have been established.

The lines of authority generally revolve around the organization's rules and regulations. Therefore, when attempting to recommend organizational change, these rules and regulations become a key factor. Far too often departments cling to outdated organizational designs and job descriptions which by their very nature provide them with the power to resist change. Rules and regulations are a highly contested terrain that is forever being negotiated, preserved, or changed. For example, with the advent of networks, e-mail and other networking technologies, individuals no longer are required to follow the chain of command ... the existing hierarchy. Now any employee can share ideas and information directly with anyone in the company; across departments and divisions; even to the CEO without going through all of the management layers that act as gatekeepers of knowledge. Many managers will perceive this as a direct threat to their authority. Those managers that perceive it as an opportunity to exchange knowledge will reap the rewards. Those that utilize a knowledge processing plan to influence the organizational structure and address issues such as these will have a strategic advantage in dealing with the territorial battles that are bound to occur.

Looking at organizational structures, it appears that many corporations are still very hierarchal in nature. This is primarily because of the way information was gathered, processed, synthesized and distributed to the decision-makers. Each division in this bureaucratic structure is required to perform these functions independently and pass the information up the corporate ladder. In rapidly changing environment in which time is a crucial element, a segmented organization that bottlenecks information through bureaucratic processes will become inefficient and unable to be competitive.

The advent of networks and relational database implies that the existing informational structure and boundaries are no longer required. Perhaps the corporate hierarchal structure and boundaries, too, will no longer be required. The concept of organizational structure as we understand it today may be radically altered as technology begins to creep into the decision-maker role through the use of artificial intelligence. Once again the ability to gather and process information will result in a shift of power.

The IS professional who develops a knowledge processing plan that clearly identifies the purpose and requirements for structural change can utilize the plan as an effective change agent to assist the organization in undertaking an orderly transformation which allows for the free flow of information as provided for by technology. From an organizational viewpoint, this can be seen as a valuable resource which assists the organization in maintaining and achieving its goals. For the IS professional, this can be an important source of quality power used as an influencer to change the organizational structure.

Influencing the Decision Processes

An ability to influence the outcome of decision-making processes is a major source of power. In a fundamental sense, one of the most basic functions of an organization is to make decisions. The quality of those decisions generally will dictate the health of that organization. Any group or individual that can exert a major influence on the decision-making processes of a company certainly should be viewed as powerful.

IS professionals, due to the very nature of technology, have a unique opportunity to influence the decision-making processes of their companies. By utilizing a knowledge processing plan that clearly incorporates the organization's business and technological processes with its goals and vision, IS professionals can be "gatekeepers of knowledge." IS professionals can influence how decisions are made and who is involved. In order to achieve this, the IS professional must get out of their domains and influence the political activity of the decision-making processes as well as the technical issues.

By emphasizing the importance of particular constraints, selecting and evaluating the alternatives on which decisions will be made, and highlighting the importance of the decision to the company, the IS professional can insist upon the establishment of a unity of purpose and vision. The IS professional must put himself out on the limb and question the wisdom of conventional practices and its decision-making processes. The knowledge processing plan provides the IS professional the tool to influence the power brokers--the decision-makers of the organization.

The Management Challenge

We have presented our knowledge processing model which uses knowledge and technology as producers of quality power, requires the development of a written knowledge processing plan based upon the concept of using technology, and uses knowledge to construct solutions to business problems. A key component of our model is the use of the plan as an influencer of power to change the organizational structure and to influence the decision processes of the business.

By utilizing the concepts as presented in our model, the IS professional has the ability to influence and control resource allocation and create a business dependency. This dependency will ensure that the IS department is viewed as a vital link to the health and success of the organization, thus guaranteeing the very existence of the IS department.

As stated earlier, organizational structure must now change to align itself with the change in knowledge and technology. In order to use our knowledge processing plan successfully, the IS department organizational structure must also change. A new skill set will be required of the personnel in the IS department. Following is a partial list of the essential activities and skills that the IS department personnel will have to demonstrate, acquire, and perform.

- Learn the process of organizational analysis
  - Learn to read situations
  - Determine power influences and constraints
  - Determine the knowledge base
  - Begin a new way of seeing and thinking
- Build information knowledge plans
  - Assist in the establishment of a knowledge plan for the IS department that fits within the corporate shared vision
- Assume the role of an inhouse consultant
- Assist other departments in creating knowledge plans
- Take the time to keep up
  - Attend workshops and conferences
  - Read journals (If you do not, the users will)
- Learn the business processes of the organization
- Move from being technical supporters to information enhancers

In a 1989 Computerworld survey, "View From the Top," a clear majority of the CEO's (64%) surveyed felt that while information systems are critical to the survival of their companies, argued with the statement that their companies are not getting the most for their information systems expenditures. In order to change this perception, it becomes imperative for the IS professional to learn to play the power game. Understanding and applying the knowledge processing plan model is an information systems management challenge. By meeting this challenge, the "viewfrom the top" of the IS professional can be changed to that of the *renaissance man* ..... "Someone who understands everything, can do everything, and is interested in everything."

5118
SUPPORT CONTRACTS

Margaret Brunner
Northern California Cancer Center
1420 Harbor Bay Parkway, Suite 260
Alameda, California 94501
415 748-6111

OVERVIEW

This paper will address the issue of hardware support contracts in a data processing shop. I will address how to find alternative support vendors as well as how to evaluate the options for the systems you want covered by a hardware support contract. I will conclude with charts comparing the prices among 5 vendors from the San Francisco bay area. This will be presented in the context of my installation which includes two HPS - a 3000/52 and a 9000/845s.

INTRODUCTION

We currently have our support contracts with HP. My organization has always carried HP contracts. HP support has been excellent. Unfortunately due to loss of revenue, I have been asked to review the maintenance cost and come up with a plan to contain or to lower the cost. This request resulted in the investigation of alternatives that produced the data for this paper.

To give you some background, we currently have an HP9000/845s with 40 megabytes of main memory, 18 ports and 2.68 gigabytes of disc space. Its peripherals include a 600 lpm dot matrix printer, a 1600/6250 bpi tape drive, and two modems to allow off site communication with the computer.

This system is principally used by our statisticians and epidemiologists.

We also have an HP3000/52 with 8 megabytes of main memory, 32 ports and 2.95 gigabytes of disc space. Its peripherals include a 1200 lpm dot matrix printer, a 1600/6250 bpi tape drive with data compression capabilities, a modem, and a statistical multiplexor with 4 ports to allow data communication and access to our San Jose site.

This system is principally used by our data entry personnel and data processing personnel.

Both these computers are linked across a local area network with 26 PCs.

Our organization gathers cancer tumor data from hospitals for research purposes. The nature of our business means that the computer is central to the operation of the organization but if the system goes down, next day repair service is acceptable. We are non profit and exist from grants and contracts, therefore, saving money is more critical than downtime.

Prior to even starting an investigation of alternatives, I would analyze the needs of the organization and the nature of the business. Have an idea of what is absolutely necessary to the organization versus what would just be nice to have. Consider whether a piece of equipment must be running and available 7 days a week, 24 hours a day or if the equipment can be down 1 to 2 days or if the equipment is non critical and can be removed from the office for a period of time. Evaluate all your equipment in these terms.

SUPPORT CONTRACT VENDORS

In order to save money on contracts, you must first investigate whether there are any alternative vendors available in your area. Support contract vendors are area specific for the most part. In order to find the vendors in your area, either attend a Regional Users Group meeting and speak with the vendors, or attend an Interex conference and speak with the vendors, or read the advertisements/classifieds in publications such as Interexpress, HP Chronicle, Interact, and SuperGroup Magazine. What you need to look for in the ads is the key word "repair" and/or "maintenance". You can also just ask other managers if there are any alternatives available in your area.

Once you have found the vendor names, contact them and let them know that you are in the market for support. Make a complete list of the hardware equipment you want on contract or, alternatively, use the current contract you have. Send (or fax) this list to the vendors you contacted, requesting bids.

Because this paper is more about methodology than findings, I decided not to publish the vendor names that I contacted. In the charts at the end of the paper that I created to compare costs among vendors, the vendors are simply labeled as A, B, C, D, and E.

SUPPORT CONTRACT QUESTIONS TO ASK

This following section addresses questions to ask of the vendor and of their references. I listed many questions. You can pick and choose among them depending on your specific needs and level of comfort in asking the questions. As an added task, besides asking questions, I would highly recommend visiting the vendor's facilities and seeing their setup.

QUESTIONS TO ASK OF THE VENDOR

QUESTIONS TO ADDRESS THE VENDOR'S RELIABILITY

- How large is your corporation?
- What distinguishes you from the competition?
- How long have you been in business?
- How many sites are you currently supporting?
- How many systems are you currently supporting?
- How many customer engineers (CE) do you have on staff?
- What are the features you pride yourself about?
- Do you have accounts my size and/or with my equipment? Who are those accounts?

What you are looking to answer with these questions is the likelihood of the vendor remaining in business. This is of particular importance if you are prepaying for their service. What you are also interested in is if they have the staff, and the experience to support your installation.

QUESTIONS TO ASSURE RECEIVING THE MOST ADVANTAGEOUS PRICE

- What is the price differential between all your support options?
- Do you provide any other type of discount for pre-payment, etc.?
- Does support need to be prepaid or will I be invoiced?
- Is the invoice monthly, quarterly, or yearly?
- What are the payment plans available?
- What are other discount opportunities?
- Do you offer any discounts in other areas if I will carry a hardware contract with you?
- Are you offering any bonus if I sign on with you?

I usually let vendors know that I am looking at various vendors. I find that this helps produce the most information and the best price and conditions. There are also many types of discount possibilities. Some offer discounts for pre-payment, some offer

discounts for 1 to 3 year contracts, some offer price freezes for up to two years, some offer services or discounts for signing with them. Investigate all possibilities to negotiate the best contract.

QUESTIONS CONCERNING PART REPLACEMENTS

- Where do you get parts from? Where is your warehouse?
- How much stock do you have on hand within a couple of hours driving distance? Where are the parts to my system warehoused?
- Do you have my hardware (specific parts) in stock?
- Will you cover all my equipment (including non HP)?
- Are complete units in stock?

These questions concern the amount of downtime you might experience if the system fails. If parts are not kept in stock locally and have to be ordered from HP or mailed from other warehouses in the state or the country, then this will add to the downtime.

QUESTIONS CONCERNING SERVICES PROVIDED

- What are your hours of coverage?
- How many site visits/preventative maintenance visits are included in the price?
- Do you do installs, deinstalls and moves? Would this be part of the contract or extra?
- What is your usual response time versus your guaranteed response time?
- Does your price cover all parts, labor and travel time?
- Is there an extra charge because of where you are located in relation to where I am located?
- Do you offer "predictive" software support?
- How many account reviews are included in the price?
- Do you do depot repair of PCs or printers(workstation products)?
- Can you rent or lease equipment or do you carry loaner equipment?
- Do you do performance consulting?
- Do you sell refurbished equipment?
- Do you work to completion or work to within contract hours?
- Do you do engineering improvement modifications as part of your standard contract?
- Do you provide remote support through remote diagnostics to determine the cause of problems?
- Do you have after hours telephone coverage?
- Do you have a maintenance site log program?

- Do you provide environmental site surveys?
- Do you have a central dispatch system? What does it consist of?
- Will I receive a written report after every service call?
- Will you support non-HP peripherals or components?

Make sure you thoroughly compare services that are of interest to you because vendors vary all over the board in what they offer. Some offer monthly account reviews, some quarterly, some semiannually, and some annually. Some will offer quarterly preventative maintenance (PM) while some do it semiannually. Some will throw in free services for you such as performance evaluation, free hardware product trials, or free deinstalls, installs, and relocation if you are their customer while some will charge for the same service or not provide it at all. Some will provide other services at a discounted rate if you carry a hardware contract with them. Some will allow you to call in HP at their cost if you are dissatisfied with their service or will pay whatever HP charges to reinstate you as an HP support customer if you are unhappy and decide to return to HP. It pays to investigate and evaluate all these services.

QUESTIONS CONCERNING THE SERVICE PERSONNEL

- Does the CE have background and training in the operating system?
- Are the CEs former HP personnel, homegrown or HP factory trained on my specific equipment?
- What is the name and background of the specific CE assigned to my site?
- Is the assigned CE trained on my machine and operating system?
- How many CEs are on staff and fully trained? How will they be trained to match my needs?
- How do your CEs stay abreast of currently released products and original equipment manufacturer (OEM) engineering improvements?

These questions are of vital concern because basically you are paying for the CE's knowledge and availability. How sharp that person is and how well trained they are will have a direct impact on how long you will be down. Also the number of CEs on staff versus the number of systems maintained is critical because you do not want all the CEs out on call if you have a hot problem. Make sure the CE assigned to you knows the operating system commands in order to perform both online and offline diagnostics. This is especially critical if you have a UNIX system.

PROCEDURAL INFORMATION

- What is the procedure for contacting the CE?
- What is your escalation management process?
- What is the after hours procedure to follow?

This is important especially if you have a very hot problem. You need to know what resources will be brought to bear on it, who will pay for these resources, and the time frame involved before the big guns are brought in.

QUESTIONS TO ASK REFERENCES

- How does the vendor compare to HP in terms of support?
- What is your level of satisfaction with this vendor?
- What is the nature of the problems that you have had with your system?
- How long have you had a contract with this vendor?
- Is there anything that really stands out in terms of pros or cons?
- What type of contract do you carry with this vendor?
- Did the vendor deliver the service as contracted (ie account reviews, environmental site reviews, PMs, written reports, remote diagnostics, OEM engineering improvements, etc.)?
- Did you look at other support vendors?
- Why did you choose this vendor?
- Who else did you contract with in the past? Why did you drop that vendor?

Be sure to ask the vendor for current references with setups similar to your own. Be sure to spend the time and talk with the references given. References will provide you with invaluable information.

ITEMS OF CONCERN

The following paragraphs detail some items that caught my attention and might not be real obvious to everyone.

Check out all statements made by the company because they are not always accurate. Either speak with their references or call HP depending on the nature of the statement. For example, Company B states in their brochure "...regularly schedules both on-the-job and off-site training sessions to either cross-train in currently released equipment or to become proficient in newly released products. For newly released equipment we utilize classes provided

by the OEM." HP on the other hand told me they will provide training courses on older equipment but not on newly released equipment such as the newer disc drives that I have or the 1200 lpm printer that I have or the HP9000/845. HP will also provide Hardware Support Subscription Services to anyone on older equipment such as 2564 line printers, the 7980 tape drives, the series 52, etc.; but not on newer pieces of equipment such as the 2566 line printer, the series 845, or the series 6000 disc drives. If improvements come along on these units, HP does not offer them outside.

Be real specific on inventory levels, especially of newer equipment. Vendors can provide support at a real competitive price because they can acquire older parts and equipment inexpensively. It is far more expensive to stock parts for newer equipment. For example, the processor board on the 845 costs around $50,000. When you visit the vendor's facilities, ask to see the specific parts in stock that you are interested in, if this is of concern to you.

Some software services are only available if you carry a hardware contract with that vendor. For example, HP will offer NETASSURE only if you carry the hardware contract for that machine with them. Be as concerned with what you will not receive from the vendor that you want or currently have as well as what you will receive from the vendor.

Double check the quotes coming back to you because the vendors make mistakes on quantity and part numbers. They might also inadvertently drop a part from the contract that you want on contract. All these errors affect the total and can lead to erroneous comparisons.

SUPPORT CONTRACT COSTS

The 4 charts at the end of the paper detail the system components for both the 52 and the 845 and show the price differentials among the 5 companies I looked at within the San Francisco bay area. As you will be able to see there is a great deal of difference among the companies. This applies not only to the total amount but also applies at the component level. Rankings among vendors also shift depending whether you are looking at a full service support contract or a next day service support contract.

In the charts, I decided only to show two of the alternatives available which are Full Service Support and Next Day Service Support. Full Service Support is Monday through Friday 8:00 am through 9:00 pm coverage with a 4 hour response time except for company D whose hours are Monday through Friday 8 am through 5 pm.

Next Day Service Support is Monday through Friday 8:00 am through 5:00 pm coverage. There are other types of coverage available such as 24 hour, 7 days a week coverage.

For my purposes, less extensive coverage was fine because we do not run a 24 hour 7 days a week shop and we can afford downtime on our computers. Please note that if you go with next day service contract, you can if necessary have same day service if you request it on an as needed basis for a flat uplift charge.

Prices shown for vendors B through E on the charts are based on quarterly prepaid support. But the prices shown do not tell the whole picture. There are many other discount options available to reduce the price. Each vendor has his own flavor. The following is a list of some of the options available depending on the vendor:

- .10% less for a 12 month pre-paid contract
- .1 month free for a 12 month pre-paid contract
- .5% less for a 3 year contract
- .Current price frozen for 2 years
- .1 month free for signing on with them
- .3 month to 5 year contracts are also available

CONCLUSION

There are a broad range of prices and a variety of options and ways to save. Be careful in your study and comparison of vendors and consider all services offered versus only the cost of the package. It might be unwise to go with the cheapest, if the service offered is also minimal. Go for the best price offered for the type of service you desire. This might not be the cheapest, but should be the one that will offer you the most overall for the least amount.

One last thing to keep in mind is that you can split systems up among different vendors if the cost and service warrants it, especially if different systems have different needs (i.e., one is a production system and the other a development system or, in my case, one is a UNIX system while the other is an MPE system.) You do not need to place all your systems with one vendor or you can just place one system with a vendor to test out their service before moving all your systems over.

Chart 1 - SAME DAY SUPPORT FOR THE 52

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES FOR THE HP3000/52 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
2	.General I/O Channel 30079A	32.00	16.00	15.00	22.00	10.00
1	.Series 40 Front 30170K	18.00	12.00	9.00	9.00	0.00
1	.Series 40 Power 30170P	14.00	7.00	3.75	5.00	0.00
1	.58 Memory Controller 30172A	0.00	0.00	7.50	8.00	0.00
1	.Basic Computer Module 30477B	217.00	168.00	89.25	105.00	100.00
2	.4MB memory array 30479A	0.00	0.00	0.00	0.00	0.00
1	.Power Supply 63909F	23.00	15.00	11.25	14.00	0.00
4	.ADCC-MAIN 30018A	76.00	20.00	27.00	36.00	16.00
4	.ADCC-EXTENDER 30019A	76.00	20.00	27.00	36.00	16.00
3	.571MB Disk Drive 7937H	165.00	120.00	81.00	114.00	90.00
1	.571MB Disk Dr. w/Cache 3937XP	55.00	42.00	28.00	39.00	35.00
1	.670MB HP-IB Disk Drive C2203A	25.00	20.00	18.00	18.00	30.00
1	.Dot Matrix printer 2934A	31.00	12.00	18.00	15.00	25.00
1	.1200 lpm printer 2566C	237.00	190.00	146.25	178.00	125.00
1	.6250XC/1600 tape drive 7980XC	57.00	35.00	45.00	33.00	40.00
1	.Thinlan 3000/V link 30240A	0.00	3.00	3.00	2.00	4.00
2	.X.25 Multiplexer 2335A	82.00	40.00	30.00	30.00	40.00
2	.Codex 2640 Modem 32066A	68.00	56.00	4.00	30.00	40.00
	TOTAL	1176.00	776.00	563.00	694.00	571.00

Chart 2 - SAME DAY SUPPORT FOR THE 845

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES FOR THE HP9000/845 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
1	.HP 9000 Model 845 SPU A1608X	383.00	293.76	240.00	300.00	250.00
3	.16MB ECC RAM Board A1037A	0.00	0.00	0.00	0.00	0.00
1	.HP-CIO HP-IB 27110B	4.00	2.88	3.00	2.00	4.00
1	.S800 CIO Fiberoptic 27111A	5.00	3.84	3.00	3.00	5.00
1	.6 Channel MUX 27140A	7.00	9.60	4.00	4.00	0.00
2	.Async CIO 6-Channel Mux 98196A	14.00	7.68	12.00	10.00	14.00
1	.Battery Backup Unit A1014M	29.00	21.50	13.00	17.00	20.00
1	.LAN 9000 Series 800 91786B	9.00	4.80	15.00	7.00	5.00
1	.THINMAU 28641A	4.00	1.92	5.00	5.00	3.00
1	.6250/1600 tape drive 7980A	57.00	33.60	43.50	44.00	35.00
2	.Model 1.34mb FL Disk C2204A	74.00	55.68	54.00	54.00	60.00
1	.HP 700/92 Terminal C1001G	7.00	3.84	6.00	4.00	5.00
1	.Line Printer 600 LPM 2564B	112.00	72.00	56.25	60.00	60.00
1	.8-pen Graphics Plotter 7550A	40.00	19.20	24.00	24.00	20.00
1	.Laserjet III Printer 33449A	40.00	32.64	28.00	29.00	20.00
	TOTAL	785.00	562.94	506.75	563.00	501.00

Chart 3 - NEXT DAY SUPPORT FOR THE 52

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES FOR THE HP3000/52 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
2	.General I/O Channel 30079A	28.00	12.80	13.50	19.80	8.00
1	.Series 40 Front 30170K	9.00	9.60	8.10	8.10	0.00
1	.Series 40 Power 30170P	7.00	5.60	3.38	4.50	0.00
1	.58 Memory Controller 30172A	0.00	0.00	6.75	7.20	0.00
1	.Basic Computer Module 30477B	174.00	134.40	80.33	94.50	80.00
2	.4MB memory array 30479A	0.00	0.00	0.00	0.00	0.00
1	.Power Supply 63909F	19.00	12.00	10.13	12.60	0.00
4	.ADCC-MAIN 30018A	60.00	16.00	24.30	32.40	12.80
4	.ADCC-EXTENDER 30019A	60.00	16.00	24.30	32.40	12.80
3	.571MB Disk Drive 7937H	129.00	96.00	72.90	102.60	72.00
1	.571MB Disk Dr. w/Cache 3937XP	43.00	33.60	25.20	35.10	28.00
1	.670MB HP-IB Disk Drive C2203A	20.00	16.00	16.20	16.20	24.00
1	.Dot Matrix printer 2934A	16.00	9.60	16.20	13.50	20.00
1	.1200 lpm printer 2566C	190.00	152.00	131.65	160.20	100.00
1	.6250XC/1600 tape drive 7980XC	46.00	28.00	40.50	29.70	32.00
1	.Thinlan 3000/V link 30240A	0.00	2.40	2.70	1.80	3.20
2	.X.25 Multiplexer 2335A	46.00	32.00	27.00	27.00	32.00
2	.Codex 2640 Modem 32066A	58.00	44.80	3.60	27.00	32.00
	TOTAL	905.00	620.80	506.74	624.60	456.80

Chart 4 - NEXT DAY SUPPORT FOR THE 845

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES FOR THE HP9000/845 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
1	.HP 9000 Model 845 SPU A1608X	306.00	244.80	216.00	270.00	200.00
3	.16MB ECC RAM Board A1037A	0.00	0.00	0.00	0.00	0.00
1	.HP-CIO HP-IB 27110B	3.00	2.40	2.70	1.80	3.20
1	.S800 CIO Fiberoptic 27111A	4.00	3.20	2.70	2.70	4.00
1	.6 Channel MUX 27140A	5.00	8.00	3.60	3.60	0.00
2	.Async CIO 6-Channel Mux 98196A	10.00	6.40	10.80	9.00	11.20
1	.Battery Backup Unit A1014M	23.00	17.92	11.70	15.30	16.00
1	.LAN 9000 Series 800 91786B	7.00	4.00	13.50	6.30	0.00
1	.THINMAU 28641A	2.00	1.60	4.50	4.50	2.40
1	.6250/1600 tape drive 7980A	46.00	28.00	39.15	39.60	28.00
2	.Model 1.34mb FL Disk C2204A	60.00	46.40	48.60	48.60	48.00
1	.HP 700/92 Terminal C1001G	4.00	3.20	5.40	3.60	4.00
1	.Line Printer 600 LPM 2564B	90.00	60.00	50.63	54.00	48.00
1	.8-pen Graphics Plotter 7550A	23.00	16.00	21.60	21.60	16.00
1	.Laserjet III Printer 33449A	22.00	27.20	25.20	26.10	20.00
	TOTAL	605.00	469.12	456.08	506.70	400.80

SOFTWARE ENGINEERING AND CORPORATE GROWTH

by
Julian R. Estrada C.
Carvajal S.A.
Calle 29N No. 6A-40, Cali, Valle, Colombia
Tel : 675011
Fax : 616581

Total quality as a business strategy for continuos improvement of all activities is, without doubt, one of the key factors for maintaining and enhancing a company's share in the dynamic markets that it serves, by bringing total customer satisfaction.

According to this new approach, Carvajal's MIS Department gathered its forces to define an appropriate procedure for software development in the various computing platforms of the organization. The main goal : To meet the needs of the in-house customer.

In order to achive this goal, methods and standards were developed in the following areas :

1. Software development
2. Production environment
3. Computing resources management

These methodologies were designed to meet the information needs associated with corporate growth, bearing in mind that the mission of MIS consists in providing the corporation with a national and international information service infrastructure wich will contribute to improve efficiency, management control and decision-making. It was also agreed that the methodologies would be the basis for achieving :

Clear criteria for the design and management of distributed information systems, considering that informatics is leveling-out organizational structures. This means that horizontal relationships, as opposed to vertical relationships, are the ones which will eventually guide the company or, in other words, that decision-making is the result of consensus. Furthermore, there are many instances in wich the geographic distribution of the decision-making group is such that requires fast, reliable, appropriate flow and processing of information by mean of distributed information systems. It is clear that in order to accomplish this aim it is necessary to have an adequate communications infrastructure which can provide such services.

A uniform and technically prepared information managment policy designed to ensure that adecuates solutions will be given to all tecnological and operational problems relating to electronic data transfer, software portability and referential integrity in each of the corporation's computer environments. This is especially important today, given the globalization of markets, technologies and executive leadership.

Information systems designed to provide appropriate and reliable on-line access to information to allow a modern business to compete promptly and effectively in today's dynamic and fast-changing markets.

In view of the above, MIS must become the corporate leader in the use of information resources. To do so, it must have, among other things, a Software Engineering Standars Guide wich meets the demands mentioned above. The Guide will serve to train new people and will be the basis for moving ahead withing the ever-changing world of software.

THE PROJECT

MIS Department surveyed 170 users from the most representative sectors of the company, to establish their level of satisfaction with the following services :

DEVELOPMENT SERVICES

* Attention to users information needs
* Information Systems development and maintenance
* Development of human resources in systems engineering
* Informations on modifications made to active informations systems
* User training

OPERATING SERVICES

* Integrated computer operations
* Printing services
* Disk space management
* Back-up and contingency options
* Peak/ closing time support
* Telecommunications

RESEARCH AND SUPPORT

* Advice on acquiring new equipment
* User final tools and Corporate support
* Support for micros
* Support for telecommunications

Results showed that users were not satisfied with several of the services listed above. The common denominator for complaints was the lack of a uniform and efficient methodology in developing and handling software for the various computing platforms in the company (see figure 1).

PRIVATE PACKAGE NETWORK

CARVAJAL INVERSIONES S.A.

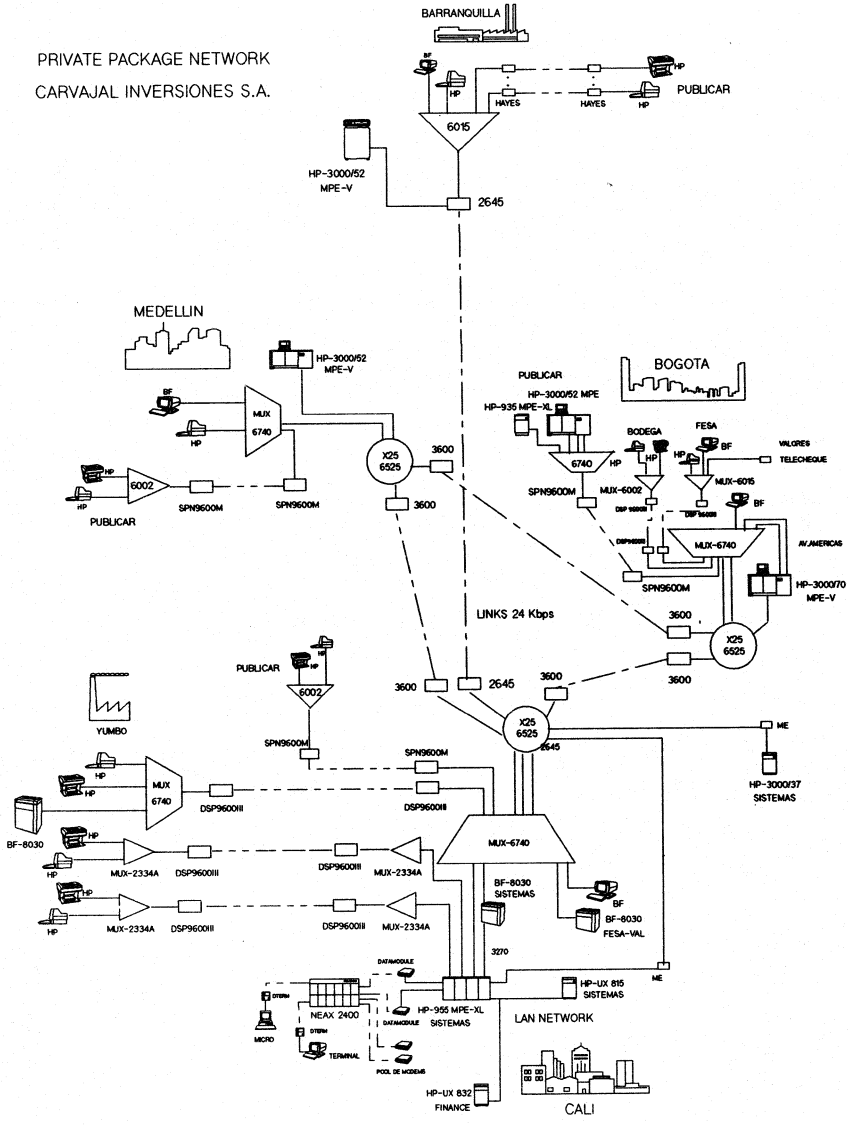


FIGURE 1

OBSERVATION OF THE PROBLEM

The existing methodology was developed for Information Systems in various third-generation languages. Analysis and design resources are therefore limited to this situation. The development cycle has in the past been as follows :

1. Preliminary study
2. Analysis and Design
3. Programming
4. Testing
5. Manuals
6. Implementation
7. Additive maintenance
8. Corrective maintenance
9. User training
10 Operation

The level of maintenance (additive and corrective) was 45% approximately and transmission errors for new software versions and data interchange were numerous. Another problem was with the low level of documentation found with the informations systems active then, leading to high programmer dependence.

Furthermore, new technologies such as switching package network, RDBs, Fourth Generation Lenguages, Open Operational Systems and so on required a special treatment.

ANALYSIS

The problem was analyzed in detail at this stage. The purpose was to discover factors which would have a high impact on the project's implementation, and define the order of priorities for action. Itwas decided that a standards document should firts be drawn up, to be applied to the development of information systems and optimun computer resource management. This would be the first step in the process of implementing a software engineering methodology associated with the growth of the company.

THE METHODOLOGY

Development Standards are a set of techniques for structuring and rationalizing the process of information systems development. They are intended to make systems development predictable, sustainable and suitable to satisfy all the purposes for wich they are created.

The development of an information system has the following stages :

1. Strategy
2. Analysis and Logical Design
3. Construction (Physical Design and Programming)
4. Testing
5. Implementation

The following is a brief description of each of these stages and an outline of their content

1. STRATEGY

This stage defines feasibility of the project, and assigns the resources required to implement it.

Activities are as follows :

Objetives

This is a brief description of the contribution of the information system to the improvement of major users' performance.

Working groups

This section defines the Users and MIS personnel who will form the working groups, they are constituted by :

USERS

. Executive in charge
. Steering group
. Leader user
. Accountable users
. Advisory users

MIS

. Area Director
. Development team

Proposal Diagrams

This is a set of graphic techniques for modelling the proposed system. Examples are context diagram, global entity relationship, functional break-down diagrams, data flow diagram level 0.

Organizational Implications

A description of the organizational implications which the new information system will entail in terms of policy and procedures, as well as jobs and functions.

Advantages Analysis

This analysis measures the overall benefit of the new information system compared to any existing system. The tools used include Effectiveness indices and Information obtainment table.

Alternatives

Alternative possible solutions are analyzed, together with the cost structures and implementation schedules.

Presentation

At this point, the final document is presented and formally validated with the working groups.

Journal

The advantages and disadvantages of the proposed methodology are noted, with the recommendation for developing of the information system selected.

2. ANALYSIS AND LOGICAL DESIGN

Analysis

Analysis is the study and appreciation of the functions of a given area, needed to design an information system. It is based on :

. Data Flow Diagram definition
. Data Dictionary
. Process Description

Logical Design

Logical Design is the logical presentation of system data in terms of entities and relationships. This should be a logical model independent of its construction context.

The following activities define a logical model proposal based on structured specification of the system given in the analysis.

. Model entity relationship definition.
. Definition of relationships in third normal form.
. Validation of structured specification and logical design with the leader, accountable and advisory users.
. Presentation of DFD and model entity relationship with the steering group and executive chief user.
. Sumary of input-output requirements.
. Journal.

3. CONSTRUCTION

Physical Design

A detailed specification in a defined computing environment based on requirements of analysis and logical design for the information system. Activities are as follows :

. Definition of the computing environment.
. Design of data structure.
. Menu and user structures.
. Initial loading specifications.
. Process specifications.
. Definition of programming schedule.
. Journal.

Programming

This activity allows for the development of the set of programs defined in Physical Design in accordance with the programming schedule defined.

A program means a logical and coherent codification for a set of requirements through programming languages. Both Physical Design and Programming are supported by a set of standards whose final purpose is to optimize computer resources and unify information management in various computing environments. The standards are:

* General standards for use.
* Programming language standards.

The general standards are not limited by the computing environment. They are, among others :

-. Labeling rules : For constructing the names of variables, tables, files and Data Bases.

-. Menu structures : The main menu is composed of the system's modules. Second level is composed of generic options like Maintanance, Process, Queries and Reports. Third-level menus are defined by each module option.

The main menu structure is :

1. Module 01
2. Module 02
   :
   :
n. Module n
H. Help
E. End

The first submenu structure is :

1. Maintenance
2. Process
3. Queries
4. Reports
E. End

The second submenu structure is :

1. Option 01
2. Option 02
   :
   :
n. Option n
H. Help
E. End

The next level submenus mantain the previous structure.

-. Security : For logon access, control and permisions to information, logs on Data bases, control numbers to avoid security violations in critical files.

-. Data structures : Design criteria for data, depending on access mode (serial, indexed, keyed), type of data structure (network, relational, files alone), location (local or distributed).

The programming language standards provide rules for each development language (Speedware, Informix, Pascal) and follow guidelines for :

-. Labeling appropiate for each language (name of actions, libraries, functions).
-. General program requirements.
-. Optimization of computer resources (disk, memory, cpu).
-. Portability guides.
-. Data input design (how the information must be gathered).
-. Data ouput design (how to show the processed information).
-. Program utilities and special language functions.
-. Help levels.

4. TESTING

The testing stage checks that user specifications have been met, and that the use of resources is efficient. The following activities are involved in defining the test schedule :

. Selection of user group to conduct test.
. Individual testing of programs, tasks and modules (addition, modify, query and delete modes in managment programs).
. Interface test with other information systems.
. Global test with real data cheking the results with the user specifications. This real data must be loaded with load software made during the construction stage.
. Acceptance test (this test checks if the information system meets the objetives for wich it was conceived during the strategy stage, the general performance and the coherence of the input-output information).
. Validate documentation (On-Line and System Documentation).
. Journal.

5. IMPLEMENTATION

Implementation begins after formal acceptance tests have been conducted by the users group in the testing stage.

Implementation involves :

. Definition of the schedule to execute the information loading process.
. Review of hardware installations (Cpus, Disc Space, Tape Units, Terminals, Printers).
. Review of software requirements (Data bases, communication protocols, run time versions in 4GL software and so on).
. Definition of the Implementation guide (Users creation on security software, D. Base capacities, connectivities review with other information systems and so on).
. Definition of Back-up procedures.
. Journal.

DOCUMENTATION

Documentation is not considered to be an additional stage of the development process, since it is created parallel with each stage. The user finally receives two types of documentation :

* On-Line documentation
* System documentation

On-Line documentation is inserted at the following levels :

1. Main Menu : General information about the system
2. Submenu : Operational information about the module
3. Menu option : Purpose and management of the menu option
4. Field entry : Meaning of field, ranges allowed, mask

Programs are documented inside source text and with the analysis and logical design stage and construction stage constitute the programming documentation.

System documentation is printed, and contains :

1. Introducction
2. Strategy stage
3. Operations guide
4. Implementation guide
5. Journal and scope notes

VERIFY

The new methodology has proved effective as a means of unifying and improving quality and efficiency of information systems. Customer satisfaction has been increased whit through participation in the development process. Additionally, the projects are developed in accordance with strategic planning of each user area, leading to better MIS effectiveness.

INSTITUTIONALIZATION

System construction needs continuous review, and regular meetings have been organized for those in projects using the various programming languages (programming clubs). There is also a high-level working group engaged in reviewing and updating the overall methodology.

CONCLUSIONS

A growing company has ever-increasing information volumes and needs at all levels of its hierarchy. There must therefore be clear policies to guide technological development and allow information to be used effectively. Decision-making must be made as easy as posible by the way in wich information is processed and data is organized. This is the guiding principle wich MIS must use in its applications and developments in software engineering.

In this highly technological area of activity, the human factor is a fundamental consideration-without it, all the standards and rules set by the software engineers would be no more than interesting ideas. Total Quality should therefore be a part of all thinking about the growth of the company; and the methodology used should be the best possible tool for achieving it.

The project for development standards must be dynamic and constantly updated to match advances in technology and take advantage of experience in application.

TITLE: To FAX or not to FAX

AUTHOR: Tom Renz

Martin Marietta Data Systems

P.O. Box 179
M/S A16330

Denver, CO 80201

(303) 790-3671

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 5120

Paper 5121

Shared data: Understanding it and using it correctly

Dennis R. Werner

Laserscope

3052 Orchard Drive

San Jose, CA 95134

(408) 943-0636

Abstract:

Hard: Understanding data that you have not created, but which you must use.
Harder: Creating solutions using existing data (which may have been created by someone else on a machine DIFFERENT from yours).
Hardest: Documenting data so that others can understand and use it easily and correctly.

Idea: "A picture is worth a thousand words."

Solution: Create a "picture" of your data to use in programming, documentation, education, communication, and working with users.

Example: One of HP's largest internal functions is electronic distribution of internal data: orders, shipments, customer information, etc. to/from almost every division and office around the world. Data is created on IBM mainframes, HP3000s, workstations, and PCs. Data is used on IBM mainframes, HP3000s, workstations, and PCs. One way that HP programmers who use some of this data are kept up-to-date on changes to the structure of these data files is by use of "data pictures."

This same solution can be used by your own organization.

This paper describes the characteristics of data and the usual presentation of those characteristics. It then describes how a "data picture" works and looks: features, designations, etc., and explains some of the reasons why such a picture communicates data characteristics more readily than alternate methods.

Introduction

[It should be noted up front that this subject is, by nature, technical - and it will generally be discussed that way in this paper. The intended audience is people, e.g. programmers, who understand the characteristics and various representations of data.]

Consider the following points:

- Distribution and sharing of data across networks is becoming more common every day. Applications running on different machines - possibly written in different languages - are accessing the same data. Somehow programmers and data administrators need to ensure that everyone is talking on an "apples to apples" (versus "apples to oranges") basis, and that they use this shared data correctly.
- Documentation is fine, but how do you document data? What about the structure of tables?
- When migrating to a new vendor, how do you verify the integrity of your data objects? How can you compare their integrity both before and after migration?
- Everyone has heard or read stories about how a "simple change" to a program caused a major catastrophe. How do you verify that "simple changes" to data definitions are correctly implemented and have the desired effect?
- With all of the languages and data construct implementations available, how do you ensure that COBOL programmers can actually communicate to your DataBase people? How do you ensure that your DataBase people can actually talk to your 'C' programmers? How do you help your managers to actually talk to the people doing the implementation?
- "Open Systems are coming!" Open systems require a greater understanding of processes and data on the part of your technical staff. How do you gain this understanding?

Everyone who has either written documentation or read extensive documentation knows that documentation can be difficult and time-consuming to produce. And, if done the "normal" way, terribly boring to read. However, I think we all recognize that documentation is important: it is something we need to use, and it is certainly helpful if it is easy to create and use.

What this paper is about is a way to make one component of documentation relatively easily, and to make it in a way which is much more usable and much more understandable - not only by ourselves - but by the people for whom we create documentation.

Who are these people? Properly speaking, documentation is not something which is written only by and for software designers and programmers. And it shouldn't be used exclusively by those people. Documentation is something that software designers and

programmers should use to share knowledge about the data, programs, or processes that they are using, that they've written, or whatever the case might be, both amongst themselves, and with the users of those systems, those programs, those data files, etc.

This paper obviously can't deal with all aspects of data documentation. Rather, discussion will be limited to the representation of data (one of the rarer of topics), particularly in such a way that users can understand it as well as the programmers.

Users can already understand the flow of data fairly well. Almost all users whom we encounter today at low and intermediate organizational levels have had some form of introduction to data processing. And as much as most programmers dislike flowcharting, the process is understood - and even doable - by virtually every user who has had an introductory data processing course. [Flowcharting is essentially the use of pictures to represent the flow of processes and data. It is almost a standard. And it is fairly easily understood and fairly well represents the processes which are portrayed by it.]

The more difficult concept of data representation is one which I believe can also be helped by the use of "data pictures."

Data, however, doesn't lend itself to flowchart-type pictures. But, I still proclaim that "a picture is worth a thousand words" when describing data - if the picture is properly constructed. The question or problem is "how do you make a picture of data?"

[To give you a peek ahead at what this paper is about, Figure 1 shows a picture of a sample data record and its component elements, i.e. a "data picture". **Note that all pictures in this paper have been manually clipped to fit in these proceedings. A normal picture occupies most of a full 7" x 10" print area and has 4 frames of 100 bytes instead of 3.**]

Sharing data across machines/architectures/languages - how it happens

Today, you will find no such thing as a "typical modern office environment." Some offices contain PCs connected to HP3000s. Some contain networks of PCs, workstations or both PCs and workstations. You'll find lots of stand-alone PCs. In some offices, you'll find combinations of HP3000s, IBM mainframes, DECs, etc. I think it's also true to say that there is no such thing as common (or true) interconnectability across all platforms.

As a result, what we are finding is that people are pushing data files back and forth (sharing data) any way they can. Sometimes they're sharing data electronically. Sometimes they are pushing files back and forth with tapes. Sometimes they use floppy disks. It's really anybody's guess what you'll find in any given office in America on any given day.

What is common among all of these offices/systems is the desire by somebody on one platform to use somebody else's data. If people are lucky, they'll find the data they

MARKETING ORDER RECORD

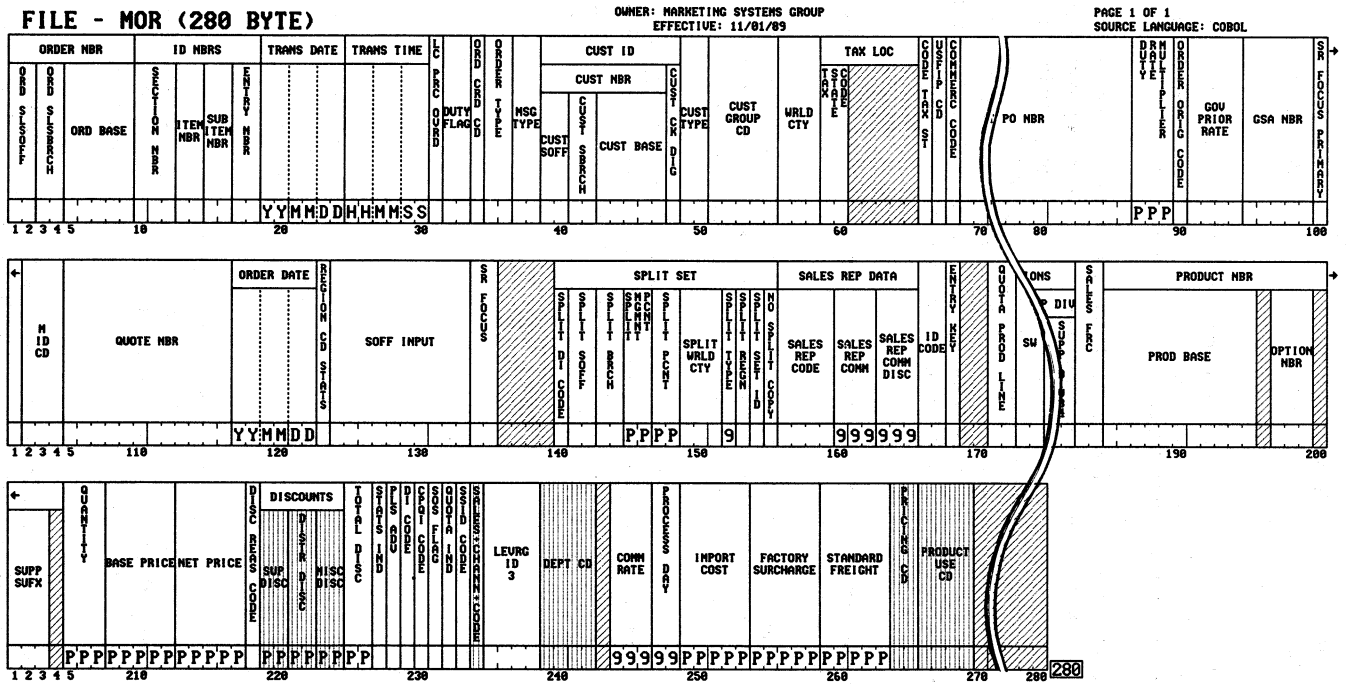


Figure 1: Sample data record 5121-4

want on their own platform, they'll have easy access to it, and there won't be a lot of translation or understanding needed. If they're lucky, they'll have the data defined in a 4GL with a dictionary, and they merely have to "use" the data. [The set of Cognos products is probably the best example of this in the HP3000 world today.] And, luckily, we're finding more and more examples of dictionaries and 4GLs in use with each passing year.

Today we're also finding a great number of people using PC applications such as Microsoft Excel. These applications are excellent at taking data produced by other systems, and using the data in some rudimentary (and not-so-rudimentary) form. I say rudimentary, because what these applications do, is recognize data as being of alpha or numeric type. They simply have the capability of allowing certain manipulations to be performed based on the type of the data item (arithmetic on numeric items, font selection on alphanumeric items, etc.). And the way that these packages present the data (e.g. on screen and onto printers) is becoming increasingly sophisticated. There's a lot of sophistication, but not really a lot of intelligence in the innards.

What's rather clever is the way these PC packages recognize data files by means of their file "extensions". The presence of one of these simple extensions specify no more than a de facto set of rules and regulations about the characteristics and organization of the data in the file. However, the recognition of these file extensions by other programs (including those of competitors' products) has proven powerful and useful to PC users everywhere. [e.g. Lotus' Ami Pro can import document files produced using Word Perfect.] And the growing number of applications using Dynamic Data Exchange (DDE) processing today is offering even more power to PC users. Unfortunately, such data files are not easily readable as "flat files."

What this paper is addressing is data which requires "a little" more intelligence. Data which has non-simple representations. Data which has relationships to other data (positional and subordinate). Data which typically requires a programmer (or program) to interpret or make extended and complicated use of.

Understanding shared data: differing needs of programmers & users

There are two types of customers of data and data representations. The first of these customers is the programmer, the person who has to "make something happen" with the data. The second type of customer is the end-user, who merely wants to "know something" about the information represented by the data. For example: Can it do something for him/her? What is the relationship necessary to support a particular function or task that they want performed?

The needs of these two customers are different. The first user/customer, the programmer, needs to know data types, offsets, subordination of one item to another, locations, etc., which are not necessary to support the needs of the second type of customer. The second type of user/customer is more concerned with the relationship of one data item to another. [This is not the "one to one" or "n to m" relationship that programmers and analysts concern themselves with. Rather, it is a question of "what

items" appear in a record, or whether or not two items of interest appear in the same record, and are therefore accessible to the customer, etc.]

Let me give an example of each kind of these two users, just to make sure the differences are clear. A programmer is the easiest example to give: somebody writing a program on a PC in the "C" language and wanting to read a data file from a floppy disk produced on a different machine. This programmer would need to know (and be able to verify) the names of the data items, the offset (from the beginning of the record or the end of the previous data item) of each new data item, the length of each data item, and the other characteristics we've already mentioned.

The programmer would then need to code the program in their chosen language ("C" in this example) on their own machine, and see if their coding produces a function capable of reading the data correctly in the matters of offset, length, type, alignment, etc. If the "picture" of the data according to how it was created doesn't match the "picture" of the data according to the programmer, the data file won't be able to be read successfully with the structure just created. Then the programmer will have to modify the structure, adding filler items, change some item lengths, change the format of some date items in his/her "picture", etc. - until they get a match with that of the data record to be read.

The second type of user can be characterized by somebody who would ask "Why can't I report ABC and XYZ?" You know the answer to the question, but how do you explain it to an end-user / customer?

Understanding shared data is sometimes a question of communication between technical people and non-technical people. Data pictures offer a way for non-technical people to communicate with technical people. They provide a way for these people to refer to data items in non-technical terms (pictures) which still convey significant technical information to programmers and analysts in a way that is mutually understood.

Shared data: some technical considerations

So what do you do if you are a programmer, if you have to read a data file, if there's no PC solution (e.g. standard file extension) available, and if a dictionary description is not available? Well, you write a program (or create a dictionary description). And if you're writing a program to read a data file, what is it that you need to know about the data elements within that file to read it successfully?

- You need to know the starting position of each element.
- You need to know the length of each element.
- You need to know the format of each element.

[Note that the name of each element is a nice-to-know, but not required piece of information.]

The format of a data element is the "type" of the element. To correctly code for this information, you may need to know some architectural information about the data representation (e.g. integer storage mechanisms). But, the three primary items listed above are the essence of what you need to know. These items are typically shown in compiler-generated program listings. But no two compilers (even on the same machine) seem to have the same format for such listings.

Just look at a simple integer as an example. If you merely think of an integer as "an integer, a count or a number," you're probably missing something important. For example, is it 2 bytes long? Is it 4? Or 8? [The length is pretty much architecture dependent if it is stored in binary form.] Is it a "simple" integer? Or is it a "double" or "double double" integer? Is the integer stored in the record in high-low or low-high form? [If you don't know what this means, don't worry about it. PCs are famous for storing integers in the reverse form of that which larger machines use.] Is it stored as a binary number, a Packed Numeric or a Display Numeric? Is there a "sign overpunch"? And another thing about integers is that sometimes they are byte-aligned and sometimes they are word-aligned (full-word or half-word or ...). Again, many aspects of integer data storage are pretty much architecture-dependent.

Common characteristics of data

Data records and their constituent data fields have some common characteristics.

- Each record typically contains more than one data field/item.
- Each data field is adjacent to the ones on either side of it. (Technically, you have to consider filler/alignment areas as data fields, also.) There are no special separator characters (as in spreadsheet data files). They are just contiguous bytes which happen to start and stop at particular points depending on the definition of the field.
- Each field has a data name associated with it which is not stored as part of the data.
- Each field has a data width.
- Each field has a particular type (storage format).
- Each field has a possible relationship to other items (either in the same or in different records).

This is typically the type of information which requires programmers to draw and process out the relationships between the various pieces of data.

Common representations of data

This leads to the problem addressed by this paper: how data is shared between such diverse computers and programming languages, and how it is understood. For example, how does an HP3000 programmer deal with data produced on an IBM

mainframe? The word sizes of the two machines are different. The HP is an ASCII machine and the IBM is an EBCDIC machine. This is not just a simple question of "how do you mount and issue a file equation for an IBM labelled tape?" That's pretty easy now: the file equation takes care of most of the hard work. The fact that the IBM produces data in a form which is fundamentally different from that of the HP3000 is a more significant problem. But it is one which is quite addressable if you know what you are doing.

The representation of the data is not really obvious to HP3000 programmers unless they've seen IBM data before - or unless they've got a picture which shows them the IBM data file and another picture which shows them how their program represents (or tries to represent) the same data. All of a sudden, they realize that "Oops! We're talking apples and oranges!" And that's the whole purpose of data pictures. With few words, key characteristics of data are shown and key differences (e.g. alignment of the fields within a record or the size of certain fields) between files on the different systems are made relatively obvious.

The more usual way of understanding and dealing with data definitions is with "data definition listings" or data dictionary listings. These listings depict one data item after another, including such items as data name, data type, picture, starting position, length, etc. But, as someone who has used representations or listings like this before, I can tell you that it is not obvious without side-by-side, line-by-line comparisons of the two to determine where problems exist, where misalignments are, or where representations are different.

An additional feature of most data item listings generated by compilers and other language processors, is some representation of "data item hierarchy" or hierarchical association. In the COBOL language, you will see these hierarchical associations shown as data element "levels" - where a 15 level is subordinate to a 10 level, for example. In other languages' compiler outputs, you might see similar associations demonstrated by subordinate levels of indentation on the items in the listing, or by designations of "group" or "structure" items. [Indentations imply the subordination of the more indented items to the lesser indented items. Groups and structures imply superordination.]

Pictorial representation of data - concept & features

So the first thing we require of successful data picturing is that we show 1) the name of the item, 2) the size (e.g. in bytes) of the item, 3) the starting location (byte offset from some reference point - typically the beginning of the record) of the item, and 4) any unusual characteristics of the item (due to architecture, for example). Somewhere along the way we'll have to also address the more "involved" possibilities of the storage format of the item (e.g. integers can be stored in binary, packed numeric, display numeric, and a few other formats).

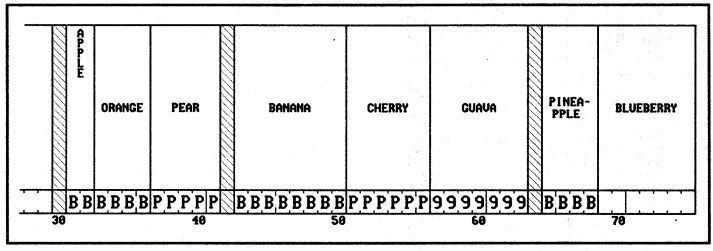
Figure 2: Sample compiler listing of numeric items

Name	Start Loc	Length	Format	Sync?
APPLE	36%	2	I1	Y
ORANGE	40%	4	I2	Y
PEAR	44%	5	COMP-3	N
BANANA	52%	8	I4	Y
CHERRY	62%	6	COMP-3	N
GUAVA	70%	7	DISP-NUM	N
PINEAPPLE	100%	4	I2	Y
BLUEBERRY	104%	7	EDIT-NUM	N

Figure 2 shows several alternate representations of integers ("Apple", "Orange", etc.). All of these representations are different, yet each could have been generated by the same compiler. They include the item name, the fact that it is an integer, its starting location and/or offset, and some attempt to describe its length.

The COBOL language - like many programming languages - uses literal "PICTURE" clauses to describe the storage attributes to be assigned to an item. Other languages, for example Cognos' QUIZ, may be a bit less obvious about how numeric data items are stored. And not all languages allow the same data types to be represented.

Figure 3: Pictorial representation of items shown in Figure 2



[Note that the picture in Figure 3 has been manually clipped to fit in these proceedings. Also note that decimal byte counts begins at 1 while octal offset counts begin at 0.]

In Figure 3, you can see a singular, non-written way of representing these same integers. This pictorial representation shows, without words, what the items look like, their length, the fact that they are integers, where they start (their offset/starting location), and any particulars about the representation method used to store the item. You'll also note that octal is not used for the starting location.

Question: Isn't Figure 2 easier to understand and more meaningful with less effort than Figure 2?

Figure 4 shows a simple data record [manually clipped to fit in these proceedings]. This data record has several data items in it. Each item has a name, a type, and a width. Some items are subordinate to others. Some items are word-aligned, some are byte-aligned, and those representations appear and stand out quite naturally.

SIMPLE RECORD TITLE

SIMPLE RECNAME

OWNER: INFO TECHNOLOGY DEPT
EFFECTIVE: 5/31/90

SOURCE LANGUAGE: COBOL

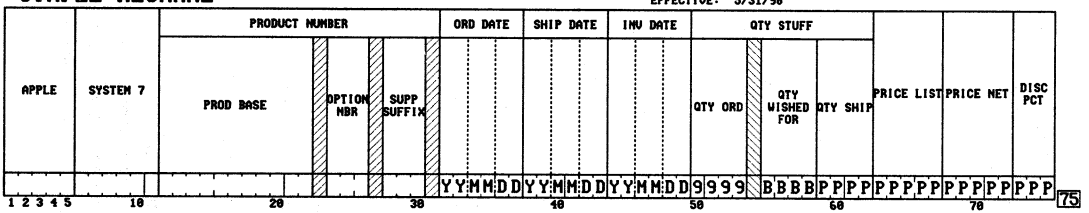


Figure 4: Simple data record

5121-10

KEY: 9 = ZONED NUM. B = BINARY NUM. P = PACKED NUM.
▨ = FILLER ▧ = SLACK / SYNC

PRINTED: 5/29/91 BY DATAPIC/PC

Looking at this picture, we examine data item #1: It has name of APPLE, begins at byte position 1, has a length of 5, and is of type alphanumeric. Data item #2 has name = SYSTEM-7, begins at 6, has a length of 6, and is of type alphanumeric. The 3rd item is a group (named PRODUCT-NUMBER) and consists of 3 subordinate items (PROD-BASE, OPTION-NBR, and SUPP-SUFFIX). [It also has 3 single byte filler fields, but I'm not counting these at this time.] The group has a length just as do its components. The next item is also a group: ORD-DATE. It is composed of 3 subordinate (nameless) items, each of which has its intended contents (YY, MM, or DD) stated. The next group, QTY-STUFF, also consists of 3 subordinate items. The first subordinate item in this group is QTY-ORD and appears to be of type display numeric (PIC 9999 for COBOL people). The next item is a 4-byte binary field named QTY-WISHED-FOR and is synchronized with a preceding "slack" byte to align on a fullword boundary. The 3rd item in the QTY-STUFF group is a PACKED NUMERIC [PIC S9(7) COMP-3 for COBOL people] field called QTY-SHIP. The last few fields shouldn't need further clarification.

Looking at the picture in Figure 4 as a whole, we see some key components. The record has a "title" [SAMPLE RECORD TITLE], a "name" [SAMPLE RECNAME], an "owner" [INFO TECHNOLOGY DEPT], an "effective date" [5/31/90], it has component items (some subordinate to others), it has footnoting to indicate some significant pieces of the picture. It also has some highlighting to indicate various aspects of the picture (such as which fields are "new" (or changed) since the last time the picture was generated). Lastly, the figure show the page number, source language, and the date the picture was printed.

Notice in this figure how the concept of data subordination (groupings) was handled. Here, the horizontal group lines clearly delineate the coverage of a group. Elements are easily recognized as belonging to their parent group. If group and item naming is done to specific standards, the group and subordinate item names lend additional meaning to the picture. [If you look at Figure 1 for a moment, you'll also note that subordinate groups are represented as a logical extension to the regular group representation.]

Why "a picture is worth a thousand words"

A lot of people contend that, as with spreadsheets being passed from one PC to another PC, you simply should not have to worry about the format of data, the structural composition of the data, etc. This is true and I agree wholeheartedly!

However, if you don't have self-describing data files and if you don't have files which are created with a set of well-defined rules as to the content and structural relationships within the file, then you do have to worry about passing that information - in some intelligent form - from the person or process which creates the file to another person or process which is going to use the file.

Until such time as self-describing files (or files which in some way carry their own structural descriptive information and therefore do not need to be described) are

available uniformly across all platforms and technologies, the process I am describing - the data picture - will continue to be one of the most viable ways of portraying this information. It can make the programmatic and logical use of data much easier.

It's nice to see a picture of file data elements, because a picture shows the primary components of any record in a method which is not wording or compiler-writer dependent. You also see, at a glance, the relationship of one item to another. And you can see the entire record and its "flavor". For example, is every numeric item in the record "packed numeric"? Or is every item a binary numeric item? And what is the total length of the data record?

But, what about data listings? Since we know the most important characteristics of each data item within a data file, we could examine how to best represent those characteristics. Then, we could rewrite all compilers and language processors to produce listings in a single, common format. Is there a problem with this? Well, yes there is.

If you look at listings and if you look at pictures and consider how the information in each is absorbed, you'll see that a picture is a much simpler way of identifying the key characteristics of the data. A picture is similarly a much simpler way of identifying discrepancies between two different records. And it is a much simpler way of determining the action needed to rectify the situation.

Look at the two record segments shown in Figure 5. If the top picture represents the structure of the data file per the creator of the file, and the bottom picture represents the structure of the file per your program, how quickly can you find the error in your program's record description? [Note how item names are not significant.] You could use the same process to verify before and after images when program depiction of record layouts are being changed. Similarly, you could use this process to synchronize two different language treatments of a data file.

Remember the user who asked "Why can't I report ABC and XYZ?" If you had a picture of the data to show your user, your answer (as a technical consultant dealing with such a customer) could be "Well look. Here's all of the data we have in our data base. Here's each of the records. Here's each of the data items within those records." You can then say: "This is the ABC record, and this is the XYZ record. As you can see, the information isn't stored in the same place, and, therefore, you can't report both at the same time without some intervening processing."

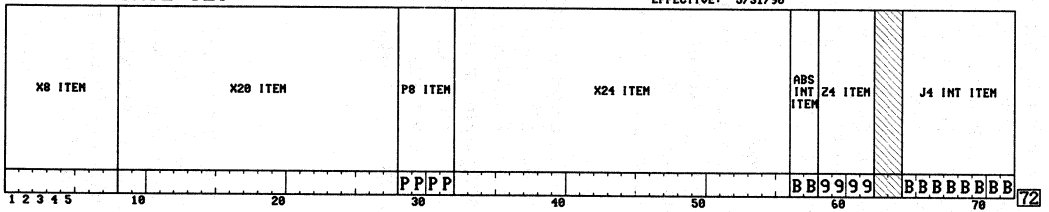
Seeing those pictures makes the user understand what you're saying a lot more than simply verbalizing or showing a schema listing or a dictionary. All to often you will also run across users who, when you show them a picture of the record, and tell them "here are all the items in it, etc.", will point to another field and say "Gee, I didn't know we were storing <u>that</u> item!" "I'd really like to have a report which shows ... in relationship to ...". You've now, by means of having a picture, given the user something <u>they</u> can understand, something they can even take away with them, return with highlighting, and say "I'd like to have a report/process which shows/manipulates ..., etc. because <u>that</u>'s what I really wanted all along."

CORPORATE SYSTEMS

TEST DETAIL SET

OWNER: INFO SYSTEMS DEPT.
EFFECTIVE: 5/31/90

PAGE 1 OF 1
SOURCE LANGUAGE: IMAGE SCHEMA



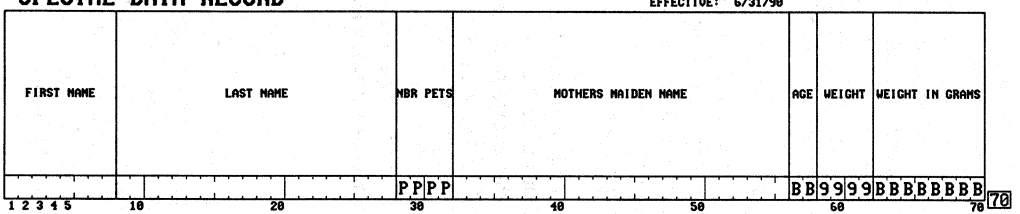
KEY: 9 = ZONED NUM. B = BINARY NUM. P = PACKED NUM.
▨ = SLACK / SYNC

LOCAL SYSTEMS

SPECIAL DATA RECORD

OWNER: DIVISION I.T. DEPT
EFFECTIVE: 6/31/90

PAGE 1 OF 1
SOURCE LANGUAGE: COBOL



KEY: 9 = ZONED NUM. B = BINARY NUM. P = PACKED NUM.

Figure 5: Comparison of 2 record segments

So why does a picture make this different from listings? First, the human eye and mind are the elements involved in how a person "sees" things. No two people quite see things the exact same way. And, it takes quite a bit of time (even though you might not realize it) for a person to translate the words in a listing into the idea behind the words. The translation function behind this part of the mental processing is the part of the reason behind the saying "a picture is worth a thousand words."

Note how one single picture in your mind represents what it could take a thousand words to describe. Therefore, if you have a common listing of each data item, its name, its starting position, its length, its type, etc., you have a great number of alphanumeric characters. All of these characters represent the single concept you are trying to portray/understand. What you are really doing is seeing a picture of the words of the representation. What follows from that is your mind translating the picture of the words of the representation back into a picture in your mind (at least that is what science thinks happens in the brain). All of this processing - to recognize and visualize the ordinate and subordinate relationships and the format of the individual items within the record - takes the human brain a fair amount of time (and "horsepower") . This is further complicated if you have groups within groups and/or repeating occurrences/instances of some components (groups or elements or both).

With a picture, the relationships between its components are implicit in the picture itself: after all, that is what a picture is all about. And with a picture, the mind skips the intermediate steps of forming some of the mental relationships because they are part of the picture: a picture cuts to the quick - goes directly to the picture of the representation without the intermediate translation steps - and removes the words from the middle of the process.

Samples like those in Figure 3 and Figure 4 shown to a fair number of programmers demonstrated this to be the case. People who have never seen a picture like this have quickly been able to discern the salient parts. [Admittedly, the footnotes help in a few cases to identify the data types involved.] Additionally and perhaps more importantly, non-technical people who see records depicted in this manner are capable of understanding, within a single record, what the data items are and what the positional and grouping relationships between those data items are. And ***that*** is what a (non-programmer) user is really looking for!

How pictures are used at HP

I worked at Hewlett-Packard Corporate Headquarters for a number of years. At HP, as you can imagine, there's a very large number of HP3000s running a great mix of programs. At the same time, Corporate headquarters has a few IBM mainframe computers (of the 3090 class at the time of this writing). There's also some DEC machines in the building, but those are pretty much restricted to datacom processing. And, over the last few years, we've seen a great number of PCs and workstations (UNIX) coming into the picture.

Order, shipment and customer information is generated on smaller HP computers around the world. It is then transmitted to Corporate for processing on the IBM mainframes. After being processed by Corporate systems, the data is transmitted back out to the smaller machines in the divisions and business units.

The function I managed was directly responsible for collecting this order, shipment and customer information from around the company and making that same information available (in both daily and historical format) to various parts of the company. We transmitted copies of the data (in tape and electronic form) to IBM users, HP3000 users, and anyone else who wanted to use the data. We also generated corporation-wide (summary and detail) reports as needed from all of the regular inputs.

Users of that data at Corporate and users of that data in the divisions stayed up-to-date on the structure of the passed data files through "change messages" and through published "data pictures" similar to those shown as examples in this paper. Each year - and with any really major update(s) to the data files - the Corporate Order Statistics group updated the pictures and accompanying information about the data files and distributed that information to all creators and users of the data.

During discussion and planning meetings, the first thing asked for and looked at were the data pictures. These pictures were sort of the de facto "bible" into the subject data under discussion. It was common to pass around highlighted copies of the pictures. It was also common to use the pictures during discussions with non-technical people (e.g. people who set policy or drove the components or quality of the data files) because the same pictures could be used and understood by all people involved.

How DataPic works - Input, Output, Language, Hardware

HP formerly produced their data pictures by manually coding the characteristics (name, starting position, width, type, etc.) of each item in a target data file into an HP3000 program which would then plot the results on an HP plotter. These plotter pictures were then photo-reduced and distributed. Because of limitations and difficulties in using this old program, I designed a PC program called DataPic which is now used at HP Corporate to produce their data pictures.

DataPic was written for programmers and data administrators who need to communicate with peers and users about the organization of their data. DataPic input consists of original language structures (e.g. a COBOL file description or Working Storage construct). It is capable of handling several languages, IMAGE schemas, and even user-written or 3rd party supplied language processors installed as "custom" languages. If a library or data base is selected as the input source, either all sets/files in the input can be pictured, or a single set/file may be specified.

Output consists of pictures drawn to a 300 dpi printer (e.g. a laser printer). Each picture shows 400 bytes of a data object per page in 4 "frames" of 100 bytes each. A single picture may span as many pages as necessary to complete the data object. Each field within the data object is labeled and annotated with either the original data name,

or with a replacement label of the user's choosing. The output picture (up to 99 copies per page) can be printed on either US letter paper or European A4 standard. Output may also be directed to a disc file for subsequent distribution or inclusion in documentation.

The picture is titled if desired, and can be tagged with particulars such as the data object's "owner", its "effective date", and can even have some of the fields overlaid with a set of hash lines to designate that they are "new or modified" since the prior version of the picture was published.

Footnotes are automatically displayed describing any special features which are part of the output picture.

Provisions are made for specifying the wordsize of target machines, or the particular architecture if alternate models (e.g. MPE-V vs MPE-XL) are available.

And hooks have been provided for "alias" item naming (via an alias file; for advancement of item naming standards within a company) and abbreviating to 3-tuples.

Also note that the currently defined process requires that the program have access to the source code (or fragment) to be depicted. If it not physically located on the same machine or accessible on an attached network, a copy of the source must be transfered to the host PC machine.

Difficulties encountered in developing DataPic

Developing most of the parts of DataPic was not terribly difficult. LaserJet programming and HP plotter programming are fairly straightforward. Language parsing and processing was easy enough. Pull-downs and pop-ups were reasonable.

The difficult part was developing "good" algorithms to automatically abbreviate and hyphenate the data names to fit them into the space available. I have to admit that before programming was complete, more than a little bit of artificial intelligence had been coded into this section of the program.

Additional difficulties were of the usual sort: user-desired extensions. As a result of user-demand, a number of "D/P Commands" were added to the language processors. These commands are included in the input language as comment records and allow specification of titles, designation of prefixes or suffixes to artificially remove from item names, designation of "New" items, alternate data item names, skipping of parts of the record, etc.

A final note about the product: the LaserJet overlays do not reproduce well on most copying machines. The reason for this is simple, but a solution is difficult: the overlays are made up of lines designed so that the printing underneath them can easily be read. Therefore, the lines are "half-tones" which are made up of individual printer dots separated by several "non-dots". These dot-lines seem to be too fine for most

copying machines to reproduce. Therefore, although the original is well-made, the copy may lose the overlay. [Depending on how the copy of this paper was made, you may see this evidenced in the various figures attached.]

A solution to the "disappearing lines" problem exists, but only for LaserJet IIP and above. In these newer machines, HP has provided for an "erase" function to "white out" a rectangular area of a drawing. This feature could be used to white out the area immediately around the letters in the item names, leaving the rest of the half-tone lines around these letters intact. The problem remains: how do you programmatically determine what version of LaserJet is attached to a machine? [You can't poll the machines the way they are designed.]

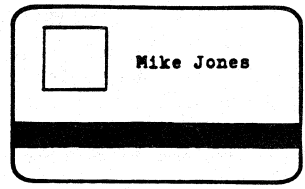
Logical extensions: aliases & other languages & dictionaries

Although hooks have been provided for "alias" item naming (via an alias file) and for abbreviation to 3-tuples, these features have not been coded to date (user demand was not great enough). Also, any other number of other languages can be added. The most popular request - and not very difficult on the surface - is the addition of a Powerhouse (Quiz) language module.

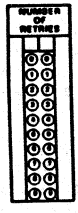
Summary

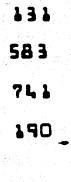
"Data pictures" have proven themselves to be a beneficial tool and means of depicting data item characteristics, both for programmers and end-users in the development and maintenance environments. They have been shown to be more popular and more understandable than the alternatives (typically various forms of listings). They have demonstrated their usefulness as technical debugging aids for successfully using shared data. They have shown themselves to be a viable communication tool between technical and non-technical (e.g. end-user) people.

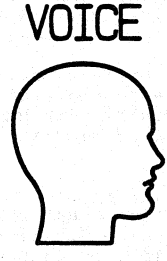
Data pictures can be generated automatically from original source language data descriptions using a PC program and attached laser printer. The PC program, DataPic, was an experiment into all of the preceding points. You can be the judge of how successful the experiment was.

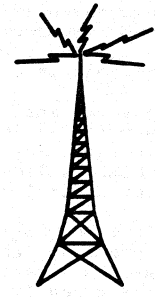


AUTOMATED DATA COLLECTION









Obtaining The Competitive Edge Through Automated Data Collection

Automated Data Collection

by
Ray Agrusti
Eagle Consulting and Development

Todays successful Industries utilize sophisticated Integrated Business Systems to be Competitive in the 1990's and beyond.

Customers are demanding that Businesses provide Superior Product/Services, with consistent High Quality, Delivered on Time, and at a Fair Market Price.

Internally, Businesses are continuously under Management Directives to be More Productive, Eliminate PaperWork, Minimize Information Errors, Improve Product/Services Quality, and Eliminate Rework and Non-Value Added Activities. These are all factors that effect overall Customer Service and Company Profitablity.

On-Line Real-Time Automated Data Collection integrated into an organizations Business Systems has repeatedly been demonstrated the Best and most Cost Effective method of achieving those goals. Business Applications ranging from Manufacturing, Distribution, Retail, Health Care, Maintenance and Insurance have all benefited from Automated Data Collection.

Let's examine todays Business Systems and identify Why Automated Data Collection provides such a Critical Component to overall Success.

All Business Systems rely on Timely, Consistent and Highly Accurate Data to be effective for an organization. Without Good Reliable Information; Procurement, Scheduling and Delivery of Product/Services tends to become ReActive and Crisis driven. The Goal is to operate in a Planned/ProActive Business Environment with all Critical Information constantly Monitored and appropriately Responded.

However, most daily Business activities consist of too many HandWritten Forms, each containing many Multi-Character Data Elements. The difficulty with HandWritten Forms are that they are Not a Timely method of Data Collection and are frequently very Error Prone.

Manual Forms typically remain Idle for a Period of Time before processing, are Mailed around the Company or even Lost. When Data is entered into the Business Systems, Much Information is frequently Illegible or Incorrect due to KeyPunch Error.

Studies have concluded that for Professional Data Entry Specialists, One Data error occures for every 300 Keystokes. If you look into your organization at the size and number of your Critical Data Elements and Multiply by the number of Business Transactions performed Daily; you will determine that Data Errors are created every Minute of every Day. Additional Errors are created as the same Information is written over and over from paper document to paper document throughout your Company.

Data Errors are insidious in their creation and always Come in Pairs. The item that needed to be updated; was not, and the item that did not require updating; was updated. Again, Industry studies have concluded that it takes 2 to 4 hours to Discover, Research and Fix each of these two Data Errors. This results in many Data Chasing/ Fixing Non-Value Added Hours spent Daily within most organizations.

The success of Automated Data Collection Systems is that they provide the methods by which On-Line Real-Time PaperLess Business Transactions may be performed. Coupling Automated Data Collection with Bar Code Technology elevates the Data Error rate to One Error every 3,000,000 Characters. Additionally, Bar Coding provides Consistent Portable Data that can flow anywhere within an organization and Repeatedly be Scanned with Virtually 100% Accuracy and Reliability.

In Summary, Automated Data Collection is the Proven and most Implemented Solution to attain the Business Goals to be More Productive, Eliminate PaperWork, Minimize Information Errors, Improve Product/Services Quality, and Eliminate ReWork and any Non-Value Added Activities.

The Seminar Session provided is a series of Successful Automated Data Collection Case Studies from the Manufacturing, Distribution, Maintenance and Insurance Industries. Discussion includes various types of Automated Data Collection Technologies and Live "Hands-On" Demonstrations. Eagle Consulting is experienced in Automated Data Collection Implementations with such companies as Hewlett-Packard, Hoechst Celanese, Westinghouse, James River Dixie and Cosmair Loreal.

You owe it to yourself and your company to explore the tremendous rewards that can be derived from integrating On-Line Automated Data Collection into your Business Systems.

Obtaining The Competitive Edge Through Automated Data Collection

OBJECTIVES

- Streamline Information Flow
- Reduce Manual Data Manipulation
- Reduce Manual Record Keeping
- On-Line Integration of Systems
- Reduce Activity Cycle-Time
- Automated Data Collection

TRANSACTIONS

- Timely
- Accurate
- Efficient

CRITERIA

- **Easy**
- **Fast**
- **Accurate**

ERROR RATES

KEYSTROKE 1/300

BAR CODE 1/3,000,000

Obtaining The Competitive Edge Through Automated Data Collection

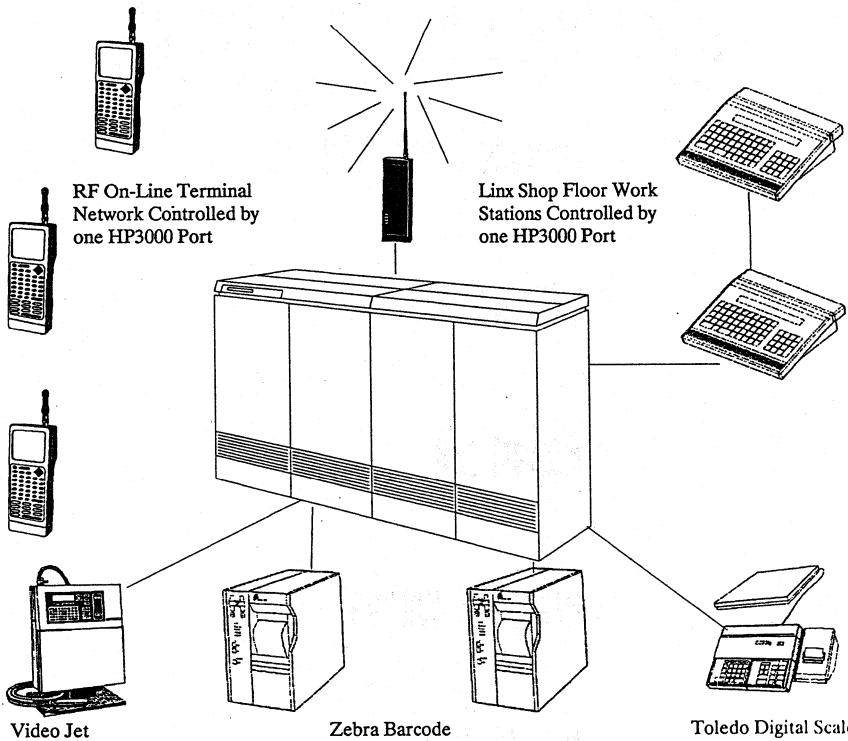
5122-7

OBJECTIVE

ERROR PREVENTION RATHER THAN ERROR DETECTION AND CORRECTION

Obtaining The Competitive Edge Through Automated Data Collection

RF EXPRESS UNIVERSAL DRIVER



RF TRANSACTIONS

- On-Line Real-Time
- Interactive
- PaperLess
- Error Preventive
- Bar Code Accuracy
- Timely "Point of Activity"
- Device Control

On-Line RF Terminals

by Ray Agrusti
Eagle Consulting & Development
Kinnelon, NJ 07405
(201) 838-5006

The timely collection of accurate data is vital to the effectiveness of today's business systems.

On-line RF terminals linked to an HP3000 processor have joined the ranks of other automated data collection devices capable of providing easy, fast and accurate data communications with business applications.

The industries of manufacturing, warehousing, distribution and retail have all discovered the flexibility and superior performance of RF terminals to satisfy their on-line portable data communications requirements.

On-line portable RF terminals provide business applications with real-time two-way interactive data communications.

A typical on-line RF system consists of a number of portable RF terminals linked via ultra-high frequency (UHF) radio waves to a single base station.

The base station, serving as an RF receiver/transmitter, plugs into a communications multiplexer that directs the data traffic "from and to" the appropriate RF terminal.

Lastly, the communications multiplexer connects via a cable to an HP3000 port. Within the HP3000 processor an RF driver program handles all the prompt generation for the terminals and updating of databases upon receiving terminal responses.

It should be noted that your organization will be required to obtain a radio license from the Federal Communications Commission (FCC) to operate an RF base station. This licensing process typically takes 60 to 90 days.

RF Data Network Components:

- ***Hand-Held RF Terminals***
- ***RF Base Station***
- ***Communications Multiplexer***
- ***Host Processor***
- ***RF Driver Program***

There are two basic approaches to implementing an on-line RF system. One configuration uses the host computer to instruct the operator of the hand-held RF terminal to execute a task thru a series of scripted commands. The Operator responds to each command with a simple "YES or NO" acknowledgment that the task has been accomplished. This technique minimizes the decision making and data collection effort required by the operator.

An example of this implementation approach is a highly structured distribution picking application. Here the finished goods item, quantity required, picking location and shipping location are clearly defined and only an indication of "task completed" is required by the computer system.

However, many business applications require a more interactive dialogue with the operator. The operator may be required to make decisions in order to accomplish the task and to collect pertinent data during the process.

In this second scenario the computer prompts the Operator for some variable-type data. The operator then selects the specific data to be captured and responds via the RF terminal. When collecting data, bar coding should be utilized whenever possible to benefit from its ease of use and high data capture accuracy.

(THIS ARTICLE APPEARED IN SUPER GROUP MAGAZINE NOVEMBER 1989)

Automated Data Collection Benefits:

- ***Reduction in Manual Data Entry***
- ***Increased Record Accuracy***
- ***Increased Productivity***
- ***Elimination of "Double Checking"***
- ***Provides Audit Trails***

Examples of this type of implementation approach are Dock Receiving and Random Storage systems. Both applications require the capture of such variable data as item number, lot number, actual quantity, and storage location. The collection of these data in response to specific computer-generated questions is integral to the application and a simple "YES or NO" acknowledgment will not suffice.

Additionally, interactive RF terminal dialogue allows the operator to be more flexible and responsive to sudden unplanned changes such as the processing of a priority request. When the response sequence or response items must deviate from that normally expected, the operator can easily indicate this condition to the computer and then be quickly redirected thru an alternate set of processing steps.

It is important to discuss the distinction between on-line RF terminal processing and batch RF terminal processing.

On-line RF terminal processing provides real-time wireless portable data communications with business applications and databases. This is far superior to batch RF terminal systems or portable PC transaction processors which may transmit data packets in an unvalidated and untimely manner.

An on-line RF terminal system will provide additional flexibility and savings by eliminating the constraints and expenses of a dedicated PC data concentrator, associated custom PC software and the myriad of host upload/download interface programs attempting to handle all possible situations.

On-Line RF Terminal Benefits:

- ***Wireless, Portable Data Communications***
- ***Paperless Transactions***
- ***Up-to-Date Database Information***
- ***Eliminate Batch Data Transfers***
- ***Eliminate "Blind" Data Transfers***
- ***Eliminate "Fear of Lost Data"***
- ***Avoid Custom PC Software***
- ***Flexibility to Respond to Changes***

In summary, on-line RF terminal systems have proven themselves in both the industrial and retail sectors.

On-line RF terminal capturing bar-coded warehouse data

RF Terminal Applications:

- ***Real-Time Receiving***
- ***Real-Time Warehousing***
- ***Real-Time WIP***
- ***Real-Time Quality Control***
- ***Real-Time Shipping***

The growing popularity of RF terminal systems can be directly attributed to their cost effectiveness and the many benefits realized through on-line data communications. You owe it to yourself and your company to explore the tremendous rewards that can be derived from integrating on-line RF terminals into your business systems.

AUTOMATIC
I.D.NEWS
FOR AUTOMATED DATA CAPTURE SYSTEMS MANAGEMENT
VOLUME 6, NO. 7 AN HITCHCOCK PUBLICATION JULY 1990

SPECIAL REPORT

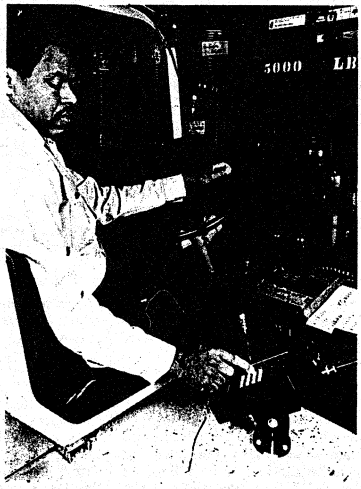
RADIO FREQUENCY DATA COMMUNICATION

MATERIAL HANDLING CHALLENGE MET

The Printing Products Division of Hoechst Celanese needed to increase efficiency and improve control in inventory tracking—a challenge complicated by the wide diversity of materials the company handles. The Somerville, NJ-based company receives and tracks 10,000-lb. coils of aluminum, 500-lb. boxes of coiled photo film and dry, granular and liquid photo-processing chemicals stored in bags and drums. In addition, the company inventories more than 5,000 end products it produces for three market segments.

Adding to the challenge was the varied storage conditions required. Certain materials have to be kept under cold conditions, others in ambient temperatures, and others are stored outside.

By integrating automated data collection with radio frequency (RF) terminals and bar coding, the Printing Products Division was able to increase record accuracy, reduce inventory and operating costs and improve expiration tracking. The improved tracking and availability of materials information will allow Hoechst to attain preferred supplier status with its customers.



RF terminals are fork-lift mounted to allow material movement transactions to be tracked at the time and place the material is moved. Employees were comfortable using the system within a couple of weeks.

"Realizing the total number of end items and the different types of data required led the company to bar coding," explains Phil Tonachio, distribution manager for receiving and shipping. Tonachio is responsible for all material movement—from raw material receipt to distribution to manufacturing—in the Printing Products Division, which has annual sales of about $120 million. The company supplies graphic arts material to printing businesses under the Enco label.

A TEAM APPROACH

Hoechst Celanese was able to determine its inventory-tracking-system requirements with the help of Eagle Consulting and Development Corporation, which designs systems for Hewlett-Packard manufacturing systems. Ray Agrusti, an MRP II and automation consultant for Eagle Consulting, defined the user requirements. With the help of the receiving department implementation team, led by Jim Conroy, distribution warehouse supervisor, Agrusti then designed the system. The system links two Hoechst Celanese warehousing facilities and a manufacturing facility. One warehouse is about a mile from the manufacturing building, while the other is about five miles farther down the road.

Other team members included Frank Arcidiacono, shipping and re-

ceiving supervisor for the manufacturing facility, and one "lead person" from each building—Jeff Adams, Bob Ruzanski and Henry Timothy. Two persons from Information Services, Manager Jeff Taylor and Frank Brown, completed the team.

Robert Cozzi, who was then distribution manager, was instrumental in the initial planning stages. When Cozzi was named manager of converting, Tonachio was promoted to distribution manager and worked with Agrusti during the installation of the system.

CONSULTING FIRM RECOMMENDATIONS

Agrusti gave the implementation team an overview of how an RF system works. "We then told Ray what we needed it to do for us. He advised us to go with a radio data network from Norand Corporation and he customized the system to provide the applications we wanted," says Jim Conroy.

Bar-coded labels, applied in receiving, provide the Printing Products Division with consistent and reliable data for the quality, manufacturing, formulation, converting and shipping processes.

Now, receiving, manufacturing and shipping processes are all on line with the company's materials-handling system, allowing the materials management department to track the receiving of raw materials, movement of materials, issue of materials, and perform cycle counting.

Sixteen Norand RT2210 radio frequency portable terminals equipped with scanners capture the data when it arrives at the receiving dock, posting the information using Eagle's RF Express application software immediately into an HP 3000 MRP II system. Simultaneously, the system prints Code 39 bar code labels, using Zebra thermal-transfer printers (Models Z-130 and Z-220 with the Zebra-Mate companion, stand-alone keyboard), which are then attached to the drums of chemicals, boxes of film and rolls of aluminum that make up the majority of raw materials used. A back-up manual keyboard allows the continuous printing of bar-coded-materials labels if the main computer is down.

The large number of end items, and wide variety of packages—corrugated boxes, plastic containers, and drums—as well as different types of storage conditions, required a label that could be applied to a variety of containers stored under various conditions. It took nearly three months of evaluating various label stocks before Hoechst Celanese selected the Standard Register thermal-transfer labels it now uses.

Distribution manager Phil Tonachio (left) oversees the RF and bar coding system with the help of Steve Vorel, president of Eagle Consulting and Development Corporation, whose company helped Hoechst Celanese select and install the radio data network.

"In an era of worldwide competition, innovative solutions to information collection help companies stay competitive," says Steve Vorel, shown here with Hoechst Celanese's lead person in receiving, Henry Timothy.

SYSTEM FACILITATES NON-STANDARD PROCESSES

The RF terminal facilitates interactive dialogue between the MRP II system and the operator, allowing the operator to send data to the system and receive instructions from it, both via real-time RF communications. It also allows the operator to be more flexible and responsive, redirecting the operator through an alternate set of processing steps when the response sequence or response items must deviate from the normal process.

Under the old system, an employee would check the incoming package list of raw materials and write up the necessary information. The package list and written information would be brought to the receiving clerk, who would then input the received data into the system. Finally, the written information would be manually input into yet another system to produce labels.

Now, employees use the portable, hand-held RF terminals to scan or key data directly into the system. The on-line RF terminal processing provides real-time wireless portable data communications between the warehouse and the data base, which has greatly increased data accuracy and ease of data entry. The RF terminals are able to pick up data errors immediately by validating all entries against the MRP II data base.

Each of the three buildings has its own base station, which is plugged into a Norand communications multiplexer. The multiplexer directs the data traffic from and to the appropriate RF terminal and connects via cable to an HP 3000 port located in one of the buildings. A single RF Express driver program in the HP 3000 generates terminal prompts and updates data bases upon receiving terminal responses.

INTERFACE WITH OTHER DEPARTMENTS

Because the RF system is only on the MRP II system, when materials are dispersed to other departments currently not running MRP II, such as Bob Cozzi's converting department, the inventory must be removed from the MRP II system. Using HP Vector personal computers and Caere wands hooked to each terminal, Hoechst Celanese employees remove the inventory by simply scanning a customized template containing bar codes for all the numbers and functions. Employees never need to key data directly into the computer, which reduces chances for error and increases operator efficiency.

The company also uses the PCs to review all the activity of specific inventory as it travels through receiving and manufacturing, and to get an overview of various categories of inventory.

USER-FRIENDLY SYSTEM MADE TRAINING EASY

Agrusti also provided training for employees. In addition to creating a user's manual, he was on site frequently during the training period.

Lead person in receiving, Henry Timothy, found it was easy to learn the new system. "Ray showed us how to operate the new equipment, and he was down here with us whenever possible. One of the biggest problems was just getting into the habit of using the system and getting used to the way it functions. Once we got the hang of it, it was easy. You can take the system and sit at your desk to do the transactions, instead of jumping from one machine to another one."

"Within a week or week and a half, they all knew how to use the system, then we had those initial people train the others," Agrusti says.

RIGHT-TO-KNOW COMPLIANCE

In addition to managing the large number of end products, varied storage conditions, and separate locations, Hoechst Celanese also needed to be able to comply with New Jersey's right-to-know laws, with OSHA (Occupational, Safety and Health administration) and ESHA (Environmental Safety Health Administration), and with the Department of Transportation.

"We had to meet not only our inventory control needs, but the requirements of the right-to-know laws. This system allows us to provide the required data to our customers, to local law enforcement and fire officials, and to the government regulatory agencies. Being a part of the chemical industry, we have an important responsibility to our community and our employees," says Tonachio.

To meet the right-to-know requirements, Hoechst Celanese now prints out New Jersey right-to-know labels using Zebra printers for all the chemical products immediately following the printing of the bar code labels. The human-readable right-to-know labels are then applied to four sides of each skid.

In addition to increased inventory accuracy, installation of the RF and bar coding system has created better relationships with some suppliers, adds Tonachio. "They see what we're doing and they try to become better customers for us. They are going into bar coding also."

AN EDGE OVER THE COMPETITION

Having the RF system gives Hoechst Celanese an edge over its competitors, concludes Tonachio. "Good inventory control, good control from an ESHA point of view, are factors customers look at when selecting preferred suppliers," says Tonachio. "We are working to bring ourselves into the twentieth century, positioning the company for growth. The bar coding and RF will give us maximum benefit.

"Using radio frequency gives the individual person on the floor immediate access to the system, even though we are spread throughout the area with quite a bit of square feet to cover."

The next step for the Printing Products Division is to begin creating bar code license plates for skids, which will eliminate the need to scan each item on the skid. The company also plans to expand the RF system to Bob Cozzi's converting area later in 1990.

"The more information we have, the better able we are to control our own destiny, and that gives us an edge," concludes Tonachio. ■

Spectacular Results With MRP II

Gisela Rank

"Everyone needs to know the direction the company is going. The next step is to teach the mechanics of MRP II implementation. Then introduce hands-on training on bar code data collection equipment."

—Tom Dowd, Sales Planning Manager, Intermec Corporation

Manufacturing Resource Planning (MRP) arrived in the industry in the late 1950s: today, running a manufacturing company "by the numbers" of precise scheduling and ordering is as important as ever. Mainframes started crunching the numbers in the 1960s. But up-to-the-minute capacity planning and shop floor control reports did not get the accurate data required for MRP II peak efficiency until the bar coded 1980s. Bar coding offers the simplest, most efficient, and most accurate data collection means available.

A realized MRP philosophy requires scheduling for customer orders, production, and purchase orders—all linked to a central computer. A Class A company meets stringent requirements: inventory, work-in-process, and finished goods have to be 95 percent accurate. Work orders must be released 95 percent on-time. Bar code is an essential link that ties all manufacturing processes to the central computer. RF technology offers a valuable supplement in heavier industries.

Class A Results

MRP requires company-wide education—and subsequent changes. Sales, marketing, order entry, distribution, and finance must integrate planning so that manufacturing has enough items stocked for product shipments as soon as orders arrive.

Under MRP II, the sales plan is tied to company objectives as well as the manufacturing master schedule. Materials keep flowing, workers keep busy, and sales/marketing is happy about the satisfied customers.

Of course, educating employees means changing some perceptions. Fear of job loss, faith in old traditions, and other misunderstandings are road blocks to success. When education and equipment revisions are accomplished, an MRP II system can be up within 18 months. Inside, in-house education is often the difference between success and successive problems. Outside training and trainers meet with the least success.

MRP and bar code must be explained in terms of the employee's role in the company. The technical know-how will be accepted in terms of the benefits to the individual first, then to the group, and, finally, to the company. The MRP II training must also include the correct educational angle for sales/marketing in addition to manufacturing.

Conclusion

MRP II makes a company more competitive—and bar code helps to deliver the essential timely and accurate information. As a management philosophy, MRP II implemented according to industry consultants gives the speed and winning edge over the slower competitor.

Computer Litigation in the 1990's
-- And How To Avoid It

Paper # 5123

Robert S. Apgood
Strategic Systems, Inc.
11050 5th Avenue NE Ste. 101
Seattle, WA 98125
(206) 362-2231

There's a shakeout happening in the industry. It's coming on fast and it's coming on strong...and certainly not everyone is going to survive.

One sure sign is the increase in the rate at which companies are seeking protection in the bankruptcy courts or failing altogether. Competition is fierce for declining budget dollars and similarly situated competitors are looking for "new" ways in which to compete. Unfortunately, one forum that they've discovered is not in the marketplace, but rather in the courts. And the reason is quite simple...it's cheaper to sue than it is to market. This is particularly true where the "shooter" is significantly larger than the "shootee." As a noted scholar in the field observes, "...the trade secret lawsuit is an unusually effective device for killing a new company. The costs can be staggering, and the attendant publicity and other restrictive effects of the lawsuit can mortally wound a new enterprise by scaring away investors, customers and employees"[1]. Indeed, one of the most astute observations he has made is that such litigation is a popular "device to eliminate or seriously cripple incipient competition."[2]

TRADE SECRETS

Just what is a "trade secret"? The Uniform Trade Secrets Act, which has been adopted by most states in the U.S., defines a trade secret in §1, ¶4 as the following:

> "Trade Secret" means information, including a formula, pattern, compilation, program, device, method, technique, or process, that:
> (i) derives independent economic value, present or potential, from not being generally known to, and not being readily ascertainable by proper means by, other persons who can obtain economic value from its disclosure or use, and
> (ii) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy.

What all this legalese really means is that such things as program source code, customer lists, algorhythms, methods of manufacturing developed in-house, design sketches, etc. could all be considered trade secrets within the definition of the law.

---

[1] The Trade Secret Lawsuit As An Anticompetitive Tool, James H. Pooley, 1983 California Law Review.

[2] Ibid

This is good. This affords legal protection to those who develop new technologies from having those technologies misappropriated by competitors. However, there needs to be a clarification here. Just because you have developed some new technique or program, you are not necessarily provided an "exclusive" right to use it. Whereas, a patent or copyright affords you legal protection from competitors copying what may be made apparent by distribution of the product of that idea (i.e., a memory board, or technical manual), trade secret protection protects only that which you have made affirmative steps to protect and maintain as a secret. If your competitor, through legal means, develops the same or similar technique, algorhythm, program, etc., that competitor would most probably be afforded the same legal protection in that his work product would be his trade secret. And, to the anguish of some manufacturers, if your competitor reverse-engineers your trade secret, it may now well be legally his trade secret, as well. This is assuming that a trade secret even exists. Case in point: The version of the Uniform Trade Secrets Act that has been adopted by California specifically states that the reverse-engineering of a product is a proper means of acquiring information. Common sense rules that once a product has been released to the public, if an examination of that product reveals the data in question, it is no longer a secret. On the other hand, the trade secret may be licensed to a competitor and still retain its trade secret quality. The main thrust, then, is that a trade secret is an idea or invention that is not necessarily unique to the owner, but is still not easily ascertainable by examination.

On the surface, this may appear to be unfair. However, the legal view of reverse-engineering activities is that society as a whole benefits when more than one manufacturer provides product offerings of similar natures. This helps eliminate monopoly situations, keeps costs to the consumers down and helps to stimulate the betterment of the products as a result of market competition. But, now we are treading into the area of economics, which is the subject of a different paper. Suffice it to say that our society has adopted stringent laws regarding the subject of monopolization and these laws are adhered to by the legal system (indeed embraced) vigorously.

So, now that you have a trade secret, how do you protect it? More importantly, how do you guarantee that it stays protected? The most important point is that the owner has taken reasonable steps to prevent the information from being disclosed. The definition of "reasonable", however, seems to lie with the trier of fact (e.g., a jury or judge). Prior to the adoption of the Uniform Trade Secrets Act, the presumption was that a trade secret existed and the burden fell on the defendant to prove that he legally acquired the trade secret. However, the standard set by the Act now demands that the plaintiff prove that he took reasonable steps to protect

the trade secret in question.

Companies that rely on trade secrets now must carry the burden of documenting that these reasonable steps were taken. This is not improper, probably due to the fact that so many of the initial trade secret misappropriation cases were born of abuse of the legal process by large companies who, when confronted with new competition by former employees, attempted to drive the new competition out of business with the onerous task of defending a costly legal battle while attempting to establish the new business at the same time. As previously stated, "reasonable" seems to be a moving target. However, some of the things that the courts have held as sufficient include:

- Physical Security
  - Controlled access to the work area
  - Security Guards
  - Restrictions on packages entering/leaving the work area
  - Restrictions on visitors and their access to work areas

- Documents
  - The maintenance and locking up of lab notebooks (you DO have your engineers keep lab notebooks, don't you?)
  - A policy regarding the maintenance of "confidential materials"
  - Confidentiality notices on documents, source code, technical manuals, etc.
  - Central Document control

- Employees
  - Employee Identification
  - Employee confidentiality agreements

While maintaining ultimate secrecy is desirable, it is close to impossible. The courts have recognized this and, in attempt to apply some substance to the "reasonable" rule, many turn to the definition:

> "While absolute secrecy is not required, there must be a substantial element of secrecy so that a third person would have difficulty in acquiring the necessary information for manufacturing the product without resorting to the use of improper means of acquiring the secret."[3]

---

[3] Clark v. Bunker (Ninth Circuit Court of Appeals 1972) 453 F.2d 1006.

EMPLOYEES

The most vulnerable spot in protecting a trade secret ironically seems to lie with the most common point of the very creation of that trade secret: the employee. Particularly in the high-technology arenas, companies attempt to employ highly talented individuals to develop products of a unique nature. Engineers, consultants, even filing clerks, in addition to being critical resources in the development and maintenance of trade secrets, are all potential areas of risk for inadvertent or intentional disclosure (read: "leaks") of confidential information.

How to protect yourself is, generally, quite straight-forward. It is quite simple to afford yourself some protection by the use of the employment process.

Preemployment: At the onset of the interview of a prospective employee, have that prospective employee execute a limited Confidential Disclosure Agreement. This immediately puts that candidate on notice that your company is serious in guarding its proprietary information. As part of the preemployment interview, impress on the candidate that the company guards its secrets vigorously.

Employment: Upon actual employment, have the new employee sign a more comprehensive agreement and supply him/her with a policy statement regarding the handling and use of company confidential documents and information.

Termination: Upon termination of the employee, whether by resignation or otherwise, review the agreement that was signed by the employee (if possible) and again explain the company's policy regarding confidential information. If there is no exit-interview process, send a letter to the employee (or enclose it with the employee's final paycheck) that covers the company's confidentiality policy. If possible, enclose a copy of the company's written policy as a reminder in either case.

CONFIDENTIAL DISCLOSURE AGREEMENTS

A Confidential Disclosure Agreement is an agreement executed between the owner of knowledge and prospective users or beneficiaries of that knowledge.

Like any other agreement, this is a contract between the discloser of the technology or information and the recipient of that

technology or information. This agreement is important in that, not only does it place the recipient of the information on legal notice that the company considers the technology or information proprietary, it also acts as further indication to the courts that the company has and maintains a policy regarding the control of that information and/or technology. Recalling our previous discussion on the burden of proof now falling on the shoulders of trade secret owner that it has taken reasonable steps to protect the secrecy of that information, this agreement becomes of significant importance in any subsequent action to protect that proprietary information.

Unless your company employs a lawyer as a member of staff, it is advisable that the creation of such an agreement for use by your company be referred to the law firm retained by your company. Since this document, alone, may well be a crucial factor in enforcing any protection, the expense in its creation is minor compared to the relative protection it affords.

As with any contract, it is prudent to be realistic regarding such things as the period of time that the information is to remain secret (the "term"). Specifying too short a term may allow a former employee the advantage of using technology or information prematurely, while too lengthy a term may result in the invalidation of the agreement by the court on the grounds of unreasonableness. Again, you must evaluate a realistic term and set that aspect of the agreement accordingly.

KNOWLEDGE VS. SPECIFIC INFORMATION

Now that you've covered all the "paper" bases, the question will arise as to what the employee can and cannot use in subsequent employment. This is the area of greatest contention in the courts. Generally, the courts will look to the skills and credentials of the employee from both a preemployment as well as post-employment perspective to determine whether or not a former employee violated a confidential disclosure agreement or misappropriated technology.

If you hired the employee for specific skills and knowledge possessed by that employee that were intimately enmeshed with the technology in question, the courts are much less inclined to side with you than they would be if the former employee had acquired knowledge of the technology as a result of disclosure in the course of his/her employment. In this vein, the courts have held that if the knowledge is so intimately intertwined with the former employees livelihood, your attempts to quash his/her use of the

knowledge are likely to be of no avail.[4] Again, the courts seem to make this determination on a case-by-case basis and no usable "rules of thumb" seem to exist.

Yes, this all does seem somewhat confusing. And, indeed, it is. The fundamental problem of trade secret law is that it is extremely difficult to determine just what does constitute a trade secret. To further compound the problem, the courts have a difficult time grasping the concepts of the computer industry, complete with its complex and evolving terminology, and are easily led down the merry path by crafty lawyers who exploit this fact, as we shall discuss more thoroughly later.

TO SUE OR NOT TO SUE

In medieval days, conflicts between two parties were settled in an arena through a process known as "trial by combat." The theory was that the will of God would intercede and the righteous would prevail. Since not all parties were equally schooled in the art of combat, representatives of the aggrieved parties ("seconds") could be appointed to do the actual combat. Since the stakes were high (i.e., someone usually died), these seconds generally commanded very high prices.

Things today are not much different. The exception being that, theoretically, no one dies. We still retain "seconds" (attorneys) and, theoretically, the righteous will prevail. Note that I use the term "theoretically." As with the process of medieval days, when justice and the law go hand-in-hand, it is purely by coincidence. What actually happens more frequently than not is that the stronger (read: more affluent) party generally prevails simply because he can afford a "bigger" second and, as such, can usually sustain the battle for a longer period of time, thereby beating down his opponent. Hence the foundation of the Golden Rule of Arts and Sciences ("The man with the gold makes all the rules").

Second only to trial by combat, the legal system today exacts its toll. Certainly, an injured party has the right to seek protection and recompense. That's what the legal system is all about. Having a forum in which to turn to settle a difference without the shedding of blood is a necessary facet of a civilized society. But, as I say, it too has its costs.

THE PROCESS

Watching Peoples Court, we see the "Plaintiff" and the "Defendant" take turns telling Wapner their respective stories. Each side

---

[4] Futurecraft Corporation v. Clary Corporation (Calif. 1962) 205 Cal.App.2d. 279

feels strongly that it is in the right and that any reasonable person (and perhaps, Wapner as well) would certainly see the folly in the opponents position. But, real life isn't like Peoples Court where, regardless of whether you win or lose, the judgement is paid out of a fund provided for that purpose by the producers of the show without contribution by the losing party. Although it can be truthfully argued that consumers (read: YOU) will ultimately pay the price for litigation of this nature, there is no guarantee at the onset of the fight that this will occur. No, in a legal battle, the losing party is "on the hook" not only for the damages awarded to the prevailing party, but to his attorney(s) and frequently the attorneys of the prevailing party as well. Losing a case of this magnitude can literally wipe out years of hard earned wealth. Particularly to the small start up company where its few principals (usually one or two) have already leveraged themselves to the hilt in establishing the business to begin with. Things such as children's college funds, years of savings, that camper you saved five years for, the second car, etc. all swirl down the legal toilet.

How does this happen? First, we need to look at the process, itself. The "Plaintiff", for reasons that we will discuss later, decides to take the "Defendant" to task. In that effort, the Plaintiff (who we will call the "shooter") hires an attorney (who we will call the "gun") to build the legal case against the "Defendant" (the "shootee"). The Gun begins an investigation within the Shooters business to see what can be found in the furtherance of the battle. Employees are interviewed. Where possible, the Shootee's products are examined (if possible, the "offending" product is purchased directly from the Shootee under false pretenses) and the initial case is compiled. If the Gun feels that there is sufficient basis to proceed, a letter is sent to the Shootee (called a "demand letter" and hereinafter referred to as the "Nastygram"). On the surface, this sounds like a trivial process. Be assured, it is not. In actuality, approximately three years of legal work are "crammed" into about two months.

The Nastygram is generally comprised of all kinds of mean, nasty, ugly references to your business practices and makes thinly veiled disparaging comments on your parentage followed by threats as to the consequences of what will happen if you continue to conduct your business. After all of this is stated, you are cordially invited to "come on down and chat it over with" the Shooter and/or his Gun. Oh, and by the way, if you don't do this in, say, the next seven days, be prepared to suffer the consequences. The consequences? In most cases, the Gun has prepared a several hundred page document to be presented to the Court requesting a "Temporary Restraining Order" (TRO) which is then followed in a couple of weeks by arguments for a "Preliminary Injunction" (PI), both of which are designed to stop you from further conducting your

business. If the claim is legitimate, or at least appears to be, the Court will usually allow both. If the claim is a "sham" or at the least, questionable, the Shooter will either not request them or the Court will not allow them, preferring to allow the matter to be settled by actual litigation and a trier of fact (either the Court acting as jury, or a panel of your "peers"...in most cases, your "peers" are six, nine or twelve people who have little better to do for the next two weeks or so).

Once the suit has been filed with the Court, the next step is a process known as "Discovery". Discovery is the legal mechanism whereby the Shooter says "show me yours" and the Shootee, if he has filed a counter-claim, gets to say "show me yours." This is the beginning of the true battle. For those of you who have never had the questionable pleasure of being involved in a lawsuit, the actual trial (like what you see on Peoples Court and Perry Mason) is the final in a long line of skirmishes before the Court. The real battle has long since been concluded. If this is all starting to sound expensive, you're catching on.

Coupled with the Discovery process is the Deposition process. In depositions, Guns for both the Shooter and Shootee get chances to ask a whole bunch of people a whole bunch of questions. This frequently involves flying around the countryside, racking up frequent flyer miles and staying in all sorts of interesting hotels in interesting places. Not only will you have the rare privilege of showing your Gun the sights of the world (all while paying him for his time...including travel time), but you will also be able to stimulate the economy by employing Court Reporters and, in many cases, attorneys in towns you've never even visited.

We've only touched on the process. But, I hope, you get the idea that it is complex and diverse in its scope.

PICKING A LAWYER

One important step that we've briefly discussed is hiring your Gun. Dependent upon the type of litigation, you will need to shop various legal firms (hardware stores) to select the Gun best suited to your needs. In high-tech legal cases, unfortunately, these Guns aren't cap pistols. They're more akin to 45 magnums like the kind Dirty Harry uses. But, like my eighth-grade shop teacher always said, "The right tool for the right job!" If you are either a Shooter or a Shootee, it is critical that you get the best Gun you can find. You will need to look for Guns that are experienced in the areas of law that are the point of your lawsuit, or in closely related areas, so that their learning curve is diminished. And, yes, you have to pay for them to learn as well (and, at $150 - $350 per hour, this adds up fast).

Once you've hired your Gun, they now take over the battle. They

talk to you. They talk to their partners. They talk to your opponents Gun. They talk to the Courts. They talk to everyone. And they charge. That's what they do for a living. And they've developed the process to MAXIMIZE the amount that they talk. And the more they talk, the more they charge. They squawk at each other. They squawk at the Court. They squawk at the Court Clerk. They squawk and they charge and they charge and they squawk. And this is the game that they play. But, while they're charging, they take every advantage they can of the system to your benefit. They confuse the Court. They confuse each other. They confuse you. And they charge. So, that's what you pay them for. And when all is said and done, you realize that the only winners were the Guns.

Sound bleak? In many respects it is. But it's the only game in town. Well...almost. Read on.

USING LITIGATION AS AN ANTI-COMPETITIVE TOOL

Using the process of litigation as an anti-competitive tool is becoming a more common practice. The financial costs can be enormous. The emotional costs can be enormous; not only are you forced into a situation where you have a full time job developing and marketing your product, but you have an additional full time job fighting the legal battle.

True, the less-than-scrupulous competitor may decide that, from a business perspective, it is less costly to litigate than to compete in the open market. However, on the other hand, the less-than-scrupulous competitor may be conducting unfair business practices, thereby forcing a smaller competitor into court to protect his business as well as the rights of his customers. In this situation, the smaller competitor faces the choice of either going out of business, or reluctantly fighting the battle. Whether Shooter or Shootee, the process is expensive. Indeed, the costs can be so prohibitive that the smaller competitor either "caves in" to the larger competitor in order to avoid the costs, or risks all in a bid to survive. The larger competitor "banks" on the smaller competitor throwing up his hands and walking away, particularly when the expense be comes so great that the smaller competitor may no longer be able to pay his Gun. This is known as the "war of attrition."

Regardless of which side of the battle you're on, one critical factor in the war is the battlefield (known as the "venue" of the court). This is of less significance when both parties are situated in the same geographic area. However, when the parties are separated by great distances, this can be a key factor in the costs. If you are forced to travel interstate to defend yourself in a suit, not only will you be required to take your Gun with you, but you may well be forced into hiring a smaller Gun ("Pistol") to

represent you in the area where the suit was filed.

This is another game the lawyers play in extracting funds from your purse. Your Gun may not be recognized by the "Bar" in the forum where the suit was filed (a "Bar" is comprised of the lawyers in an given geographical area that sets the rules of how the game is played in that area), and you must have a Bar recognized attorney representing you in the venue of the action. Therefore, you get to pay the Pistol to act as little more than a mailbox. To maximize your legal dollars, it is best to have the venue lie in a convenient court. Case in point: Recently, a large East Coast manufacturer of PC spreadsheet software won a battle against a publisher for infringement of a copyright. Another developer of similar software, anticipating that it was the next target in line for suit from the victor, proceeded to a local court and sought what is known as a "Declaratory Judgement" from that court that they hadn't infringed on the copyright in question. Since they were the first to seek a ruling from a court on the question, they picked the battlefield. Now, when the victor of the first suit elected to pursue the second company, the second company took advantage of the rule that requires all legal actions between two parties into one court. Neat, huh?

Another consideration in picking the battlefield is the rules that you have to play by in that battlefield. If you need to "discover" a great many things, you need to find the court whose rules give you the most flexibility. If, on the other hand, you are guilty of the transgressions of which you are accused, you will want a court with more stringent policies regarding discovery or with procedural rules that work to your advantage. Additionally, you will want to look for a court where the Judge has tried similar cases in the past and who has shown more sympathy with your position. All of these considerations, of course, require a great deal of research on your behalf by your Gun (read: more $$$).

STEPS YOU CAN TAKE TO AVOID PROBLEMS

The best way to avoid a legal problem is by not ever getting involved in a lawsuit. On the surface, this sounds droll. However, there are steps that you can take to minimize the possibility of lawsuits arising against you.

Arbitration

As mentioned earlier, involve your personnel department in your procedures. If you have employment contracts with your people, have the contracts constructed to contain arbitration language. In your Confidential Disclosure Agreements, provide arbitration language. In virtually every contract you write, provide arbitration language. First, let's look at a sample arbitration clause. Then, we'll discuss why this is preferable to litigation.

Sample Arbitration Clause:

"Any controversy or claim arising out of or relating to this Agreement, or the breach thereof, except as stated below, shall be settled by arbitration in the City of <your city and state> in accordance with the Rules of the American Arbitration Association, and judgement upon the award rendered by the Arbitrator(s) may be entered in any Court having jurisdiction thereof."

What this means is that the parties to the agreement/contract recognize that litigation is both expensive and time consuming. In furtherance of that recognition, the parties agree to use an alternate forum for resolving any differences that may arise out of their relationship. There are several vehicles available for this type of Alternative Dispute Resolution (ADR). Entire organizations have sprung up to act in a mediation/arbitration situation and their members are commonly former judges. However, don't get the wrong impression. These arbitrators are frequently not dottering old retired judges trying to make a buck. A large majority are judges who, tired of the lengthy legal process, are applying their knowledge and expertise to dispute resolution in forums where that experience can have almost immediate results; results that have a profound reduction in the cost to the potential litigants that they might otherwise have been subjected to in the course of traditional litigation.

If you are already in the process of litigation, or if you are forced into litigation by another party, most courts have rules that allow for, and many have rules that require, parties to attempt to resolve their differences prior to the commencement of the actual trial. In these situations, both parties can make use of a "mini-trial" of sorts whereby they make their case to an ADR panel whose purpose it is to point out the strengths and weaknesses of the arguments of the parties. While this may have the unfortunate effect of "showing your hand" to your opponent, this may well also provide you with the benefit of seeing your opponents hand. It is not uncommon for ADR to give both parties a good idea of the task at hand coupled with a realistic appraisal of their respective chances in actual trial, leading to a more timely resolution of their dispute. The overwhelming success of ADR is rapidly leading it to wide acceptance in many legal and business circles.

"Raiding" the Competition

One excellent way of increasing your chances of being sued is by "raiding" your competition of its key employees involved in the development or marketing of a competitive product. Certainly, you are entitled to attempt to acquire talent. Every company does it and this, in and of itself, is not contrary to the social well

being. What is frowned upon by the courts, however, is a definite and methodical process of soliciting and/or seeking the employing of these personnel. If, on the other hand, a key employee of your competition seeks employment with you for reasons of his/her own, don't be hesitant to seriously consider the application. In these situations, be very clear to the prospective employee that you do not wish to be the recipient of information or technology that is considered proprietary to the competitor. It helps if you can execute a letter of understanding with the new employee that you expect that employee to honor the confidentiality of his/her previous employers proprietary information just as much as you expect the same treatment of your own proprietary information in the event that the new employee leaves your company and seeks employment elsewhere.

Documentation

Whether you are an existing company, or a start up, documentation of your activities is essential both in preparing for and in the conduct of a dispute.

Institute a policy regarding the handling and protection of confidential company information. Ensure that all confidential information is noted "Confidential" and that procedures exist for the distribution of the information. This is as important for customer lists as for "secret" processes or formulas.

Have all employees execute a Confidential Disclosure Agreement in favor of the company if they are going to be recipients and/or users of proprietary information.

Have all engineers maintain lab notebooks documenting work in progress and retain filled notebooks for several years.

In any legal skirmish, success frequently can be measured by the amount of documentation that you have. The more evidence you have supporting the development of your customer lists, products, etc., the less likely a spurious litigant is going to be in prevailing in obtaining an injunction or at trial.

Earthquakes - A Strategy

Bud Beamguard
Syntex Research AR-RIS
3401 Hillview Ave.
Palo Alto, CA 94903
415-852-1416

==========

Earthquakes are a major concern for data processing managers in California and increasingly in other earthquake-prone areas, such as the Pacific Northwest, Utah, and even the Midwestern United States, where there has been much concern lately over the potential for a repeat of the massive New Madrid quake of the 19th Century. New England and South Carolina have also experienced serious earthquakes within historic times. The Loma Prieta earthquake in Northern California, 1989, resulted in extensive re-examination of many disaster recovery plans.

Earthquakes present a number of unique problems for data processing managers:

1. They are impossible to predict; there is little indication that this situation will improve despite intensive research into earthquake prediction.

2. An earthquake takes place suddenly, and without any warning. This is in contrast to other types of disasters like floods and hurricanes.

3. They occur at highly irregular intervals. In an area laced with fault zones, any of the faults may suddenly decide to move, independently of the others.

4. Earthquakes are a geological event, and a given fault will rupture only over a period measured in many years. The San Andreas fault for example is believed to produce a major quake in the San Francisco area only about once every 100 years. How much money should be spent to prepare for such a long-term problem? How do you justify it to management?

5. It is difficult to motivate people to prepare for earthquakes, and there is a large amount of denial associated such preparations.

6. Earthquakes vary enormously in power. A 5.0 earthquake will not cause any serious problems. A 7.1 like the Loma Prieta resulted in general disruption. A 8.5 like the 1906 San Francisco quake devastated the Bay Area. What level of preparation is appropriate?

7. Earthquakes are unique as a disaster event, and are not at all like fires, floods, vandalism, etc. These are localized events, but an earthquake causes far more extensive and regional disruption.

An earthquake is a tremendously disruptive event, not just for data processing centers, but for the entire region. Herein lies another set of dilemmas for the data processing manager.

1. The computer room and contents will probably come out OK after a quake (unless the building collapses).

2. Even if the computer survives, the regional infra-structure will very likely be disrupted: electrical, telephone, and transport services may not be available. In California, freeways have repeatedly demonstrated their vulnerability in earthquakes.

3. The business which the computer supports will almost certainly be badly disrupted, probably more so than the computer itself.

4. People are completely disoriented by quakes, with psychological problems which persist for many months. The DP manager is not exempt from this impact, let alone his staff.

5. Panic is general after a quake. The last thing anyone cares about is a computer. Computer recovery takes a back seat to many other concerns (children, homes).

What is an effective strategy to prepare for earthquakes? A few suggestions based on our own experiences:

1. Very important: make sure that the applications development staff and the applications users have been made aware of earthquake preparation. Get them to understand what might happen, and get them to commit to planning efforts. It is quite likely that a quake will disrupt the business more than the actual computer. Make sure that recovery priorities are established.

2. Nail down your computer room. Imagine the entire structure bucking violently: what would come loose or fall over? Raised computer room floors are often a weak point. In particular, look at tape racks, bookcases, file cabinets, and racks containing datacom gear. Unfortunately, many computer rooms are designed to be flexible and easily re-arranged: an earthquake does this very effectively. Also, overhead lighting fixtures and false ceilings have caused many injuries during earthquakes when they come loose.

   Physical preparations of this sort are actually simple, and will prevent many

problems during a quake. If you are not sure how to secure your equipment, get help.

3. Personal computers and terminals are everywhere. Are they going to take off during a quake? There are devices available to help prevent this.

4. A backup power supply for the computer room might be a good idea. But consider: if the power is out over the whole region, what will your business be like?

5. Worry about telephones. They are not going to work very well after a quake, if at all. People take them for granted. Have some sort of elementary communications strategy for the post-quake period. Establish in advance an emergency hot-line number, and distribute emergency phone lists to everyone.

6. Hot-sites in other states or regions are probably a gamble for earthquake disaster planners. How are they going to be useful should regional telecommunications be a mess? (Hot-sites might be fine as a way of preparing for non-earthquake events.)

7. Determine what your insurance policy is like. What is the deductible for earthquakes? How quickly can the vendors replace damaged equipment? (They may be in as big a mess as you are.)

8. Make sure that there is a way of getting critical passwords during emergencies (MANAGER.SYS, OPERATOR.SYS, etc.). Security issues become less important during recoveries, but unknown passwords can be a major obstacle.

9. There is probably no adequate way to prepare for the psychological trauma following an earthquake, except to be aware that it will happen. What are you going to do if your operators all suddenly vanish?

10. Set up an emergency tool-kit, reserved strictly for emergencies. Include plenty of cabling and other wire.

11. Get yourself prepared as best you can, and then don't worry about it. You really have no idea what will happen anyway, but at least you've covered the obvious.

12. Above all, don't fall into denial by saying "it won't happen to me".

It is my sincere hope that you will never experience a severe earthquake, and I hope that the 7.1 Loma Prieta will be my personal "big one". There is no way to predict whether or not your site will experience a serious earthquake, but at least there ARE things which you can do to protect yourself.

TITLE: Managing For Success - Managing UP

AUTHOR: Edward L. Bye

Infosol, Inc.

3282 S. Mountain Ave.
Suite 340

Upland, CA 91786

(714) 920-9012

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPERNO. 5127

TITLE: End User Computing - A Formula For Success

AUTHOR: Dougas Colter

Infocentre Corporation

7420 Airport Road
Suite 201

Mississauga, Ontario L4T 4E5

CANADA Phone (416) 678-1841

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 5128

MANAGING BURNOUT

Elisabeth M. Craig
Center of Excellence for Computer Applications
The University of Tennessee at Chattanooga
615 McCallie Avenue
Chattanooga, TN 37403

Abstract

If you are a computer center manager, you can expect to lose valuable employees to the burn-out syndrome. However, there are proactive steps you can take to reduce this trend. The presentation focuses on how to detect symptoms, institute preventive measures and, when necessary, employ remedial strategies. While the main emphasis is on burn-out or mental stress, the related topic of ergonomics is discussed. There are techniques which managers can successfully use to decrease physical fatigue caused by eye-strain, noise pollution and other bio-technical phenomena. Information will be provided on what experts have discovered about the optimum construction of workstations.

The presentation reviews the latest research on burn-out and ergonomics as these topics relate to data processing and other computer personnel. The management of burn-out in the 1990s requires supervisors who are knowledgeable of strategies to maintain not only a safe physical environment, but one which is mentally stimulating.

Introduction

The term "burn-out" first appeared in publications during the 1970s. The term has come to epitomize a situation in which an enthusiastic and helpful employee becomes cynical, lethargic and unable to respond to job situations. Not infrequently, the employee becomes so uncomfortable that he/she will leave the organization.

Symptoms and Effects of Burnout

Generally, burnout is considered to be a reaction to stressful work environments. Some jobs by their design place workers under constant and extreme levels of unrelieved stress. For example, Greenberg lists computer programming and related jobs as a high-stress occupation (p. 263). These jobs require continual learning, intense concentration, and adjustment to rapid technical changes. These requirements can, and do, cause both physical and emotional stress. This stress is particularly apparent among Help Desk personnel afflicted with a constant barrage of user questions and emergencies which need nearly instant responses. (Statistics indicate that 70% of Help Desk employees indicate moderate burnout symptoms within 18 months.)

Dr. Terrence Deal of Vanderbilt University states that a typical response to change (such as new software or hardware) is that employees suddenly feel incompetent and unable to respond well to challenges. This type of stress applies particularly to computer departments because employees not only have to keep up with the changes, but are on the firing-line as the "experts". Thus, managers need to be aware of indications of stress and burnout. Table 1 indicates some of the symptoms associated with burnout.

Table 1
Signs and Symptoms of Job Stress and Worker Burnout in Human Service Programs (Cerniss, p. 17)

1. High resistance to going to work every day.
2. A sense of failure
3. Anger and resentment
4. Guilt and blame
5. Discouragement and indifference
6. Negativism
7. Isolation and withdrawal
8. Feeling tired and exhausted all day
9. Frequent clock-watching
10. Great fatigue after work
11. Loss of positive feelings towards clients
12. Postponing client contacts, resisting client phone calls
13. Stereotyping clients
14. Inability to concentrate or listen to what client is saying
15. Feeling immobilized
16. Cynicism regarding clients; a blaming attitude
17. Increasingly "going by the book"
18. Sleep disorders
19. Avoiding discussion of work with colleagues
20 Self-preoccupation
21. More approving of behavior-controlled measures such as tranquilizers
22. Frequent colds and flu
23. Frequent headaches and gastrointestinal disturbances
24. Rigidity in thinking and resistance to change
25. Suspicion and paranoia
26. Excessive use of drugs
27. Marital and family conflict
28. High absenteeism

When a computer manager expects high performance he/she adds pressure to subordinates and to a job environment which already is stressful by definition. Thus, as managers we need to develop skills to reduce and manage stress, especially since it becomes more dangerous as it becomes cumulative.

Burnout due to stress does not occur suddenly. For example, Cerniss (p.17) quotes three stages in the process. In the first stage, which he terms stress, an in-balance occurs between demand of users and resources of the computer center. In the second stage there is a short-term, immediate emotional in-balance characterized by symptoms such as anxiety, tension, and fatigue which he characterizes as strain. The final stage, or what he terms defensive coping, results in changes in attitude and behavior of the employee. Often the stages are self-reinforcing because initial symptoms of one stage can cause others. However, knowledge that these stages exist can provide the manager with a method of determining where a person is in the burnout process.

Why should managers be concerned about burnout? First, this malaise is costly to a company because typically the syndrome hits long-term employees rather than newly hired personnel. The loss of experienced people is one that a company can ill afford since extensive amounts of money are invested in training, especially in the area of computer technology. Albrecht (pp. 128-132) provides an example of the cost to a company. He cites a company with 2,000 people, sales of $60 million a year, profits of $3 million, an absentee rate of 4% and a turnover rate of 5%. He extrapolates from this and other data that if it can be assumed that 2% of absentee and 2% of turnover is due to stress, then the two factors cost the company $3.5 million a year in turn-over and lost time.

Second, burnout affects the productivity of the employee and add additional stress and work to other personnel in that department. Third, since the symptoms of burnout include loss of positive feelings to users or clients, there may be a negative impact transferred to users from other departments including their perception of the computer center. Fourth, even one burnt-out employee can affect the morale of the department leading to an insidious spread of the syndrome.

Treatment

Nancy and Donald Tubesing state there not a single remedy which will reduce burnout. Indeed they state "the most potent force in the burnout equation is the suffering individual whose own internal wisdom can be activated to diagnose and alleviate the symptoms." (p. 157). This statement suggests that burnout will have to be managed at the psychological level through coping strategies though physical stress can not be ignored.

Physical Stress and Ergonomics

You can start by ensuring that your work environment is clean, cheerful and comfortable for your workers. This environment not only relates to the morale of the group but also influences physical wellness. Research is increasingly showing dangers inherent with computer equipment and provides guidelines on constructing safe workstations. The growing field of ergonomics is showing, among other things, how to protect employees and design safe workstations.

Constant noise pollution from machinery or over populated work areas can quickly induce irritation, fatigue and stress. Large offices containing many workstations create this problems; whenever possible baffling structures should be installed.

To prevent eye strain, avoid creating workstations in which the staff member has to constantly look up at a monitor. Those of your personnel who wear glasses will find it helpful to get some specifically prescribed for terminal work.

Neglect of factors such as the heights of chairs, tables and distance between terminal and user can cause physical stress. The heights of the seat should be adjustable and chairs should have wheels and if possible arm rests. Typically a keyboard should be 27 to 29 inches from the floor, but should be adjustable so that the user's elbows makes a 90 degree angle when the fingers are on the keyboard. Feet should placed comfortably on the floor.

Paul Brodeur, writing in MacWorld notes that there is increasing concern about "the potentially harmful health effects of electromagnetic emissions from display monitors" (p. 136). Research is still on-going to determine whether video-display terminals increase the risk of miscarriage. Until more research is completed, users are urged to take precautions. Thus, Brodeur suggests that personnel be at arm's length when facing a terminal. In addition, he urges that workstations should be constructed so that individuals are not within four feet of the side or back of monitors (emissions from magnetic fields are stronger from these areas). Some manufacturers of monitors now are redesigning models to lower, or in some cases eliminate, electromagnetic emissions. For example, NEC Technologies has recently done reissued its Multisync 3D monitor with a yoke which cancels the electronic emissions.

One of the maladies receiving much attention these days is Carpal Tunnel Syndrome (CTS) which effects many occupations including those associated with extensive typing on a computer. Due to constant repetitive tasks, such as striking keys on a keyboard, the syndrome can occur. It causes tingling fingers followed by numbness and sometime debilitating pain. CTS can be

treated, though in some cases there is not complete recovery of wrist and hand motion. One of the problems is that in the beginning stages the sufferer does not experience tingling in the fingers until some hours after the activity has ceased so he/she may not equate work and the symptoms of the syndrome. Positive things which you and your staff can do to avoid CTS are:

- watch each other to observe unconscious movements involving the hands and fingers such as rubbing of wrists, shaking the hands.
- rest from keyboard activities for 10 minutes each hour to provide recovery time from the constant repetitive typing.
- arrange activities so that there are breaks during the day from keyboarding.
- perform tension relieving exercises such as:
  * make a tight fist for a couple of seconds, then stretch the fingers wide out and hold in this position for at least five seconds; repeat this exercise several times.
  * raise and lower hands several times, with arms outstretched in front. Then, make circles in the air with fingers by rotating the hands ten times.

Other practical measures to reduce over-all physical stress are to suggest that employees do not take work home and that they take a full hour for lunch during which work is not discussed. In the same vein, we can encourage personnel to limit overtime and make sure that they schedule time to take vacations.

Coping Strategies

Tubesing and Tubesing state that while it is necessary to reduce physical tension it is important to teach coping strategies. Therefore, they recommend that managers help their workers learn listening skills and assertiveness, as well as "flight" skills such as daydreaming for a few minutes (p. 164-165).

There are other coping strategies which we as managers can initiate. One caveat is that managers must realize what they regard as good strategies may not be those selected by their staff. For example, Dr. Mark Mendenhall, writing in The Chattanooga Times, referred to a University of Michigan study which asked employers what they thought their workers would choose as motivators, then asked employees what they wanted. Managers guessed that job security, good wages, a clean, safe working environment and possibility of promotion would motivate employees. However, the researchers found that full appreciation for work done, the feeling of being "in" on things, sympathetic help with personnel problems and job security were the choices of the employees.

This research indicates that employees need to feel that they have some impact and control. Lack of these elements leads to a feeling of helplessness,

which creates an environment conducive to burnout. Workers want to know that their efforts are important and to see how their contributions fit into the "big picture". Helping our staff to realize their talents are valued by the department and the organization results in a creative response from the individual which is a powerful combatant to burn-out.

In the same vein, an employee should not feel a high degree of ambiguity about responsibilities and role within the department. He/she needs to know his/her identity and how it relates to the department as a whole. As managers we need to give employees task identity and also to foster supportive relations between workers so that everyone knows his/her role and how contributes to the unit.

Greenburg states that if a person feels occupational stress, he/she should discuss it with others. (p. 264). He suggests that we encourage a sense of humor, especially about stressors. We need to provide opportunities for personnel to ventilate about problems or to debrief within a positive environment. That is, the focus should be on allowing employees to talk, while channelling the discussion into developing creative and positive approaches to problems. Experimentation and innovation should be encouraged and discussed. For example, it may be helpful to discuss how difficult users can be handled and perhaps referred to several consultants rather than assigned to one consultant.

Employees should be given ample opportunity and encouragement to attend training sessions, workshops and seminars. Pines, Aronson and Kafry (p. 111) state its is a necessity for employees to have such "times out" especially for jobs which are associated with serving other people. Training not only is a solution to "time-out", but also provides a sense of self-fulfillment and attainment by giving personnel the opportunity to master new techniques. When an individual is called upon to answer questions or handle a situation for which no training has been provided, a situation conducive to burnout is created. If money is not available to send employees out-of-town, why not create in-house training in which employees take turns to cross-train each other? One advantage of internal cross-training is that participants shift areas of specialization and this causes diversification, an antidote to fatigue.

As an adjunct to training, there should be an supply of manuals as well as departmental subscriptions to journals and trade newspapers. Staff should be encouraged to spend a few hours each week reading such materials to both keep current with the latest technology and at the same time have a brief respite from pressure. Consider having your department publish a newsletter or flyer to disseminate information about the department and, equally important, provide task diversification.

We can insure that our employees receive informative training not only in dealing with equipment and software, but in coping with users who are often frightened by new technologies. Knowing that fear may cause the user to become angry and frustrated and how to handle this can be vitally important.

One idea may to be to introduce your staff to the Myers-Briggs Type Indicator. This procedure asks participants to make decisions between sets each composed of two words. The test has four scales designed to give participants insights into how people look at things, how they make decisions and how they deal with the outer world. Myers-Briggs identifies 16 types, each of which has different interests and values. The Indicator describes in a positive manner the traits associated with each type. When administered by someone familiar with Myers-Briggs, the test seems to be fun for the participants and provides valuable self-knowledge in a non-threatening manner. This informations helps not only with dealing with colleagues but in working with users. For example, it may help John, who focuses on the future, to understand why Tom, a user who tends to focus on the present and concrete information, may not see a problem in the same way.

Managers should avoid the concept that "more resources" will solve the situation. - Often obtaining more technical equipment exacerbates a situation which really calls for a solution rooted in moral support, understanding and mastering coping strategies.

Summary

Burnout is an on-going problem. Managers need to identify individuals who seem to victims and change their physical environment or help them learn coping strategies. However, this does prevent the phenomena occurring again in a different set of individuals.

In Star Trek, Mr. Spock, the perfectly logical and unemotional Vulcan causes frustration and stress for his colleagues despite their admiration for him. Our computer departments bring our employees into daily contact with a device which is logical and unemotional, while we are not. We need to recognize that stress is going to occur because of this "relationship". As managers we can look for clues that our personnel are experiencing excessive stress and implement proactive activities which can alleviate the stress before it turns to burnout.

Bibliography

Albrecht, Karl. Stress and the manager. Prentice-Hall, Englewood Cliff 1979.

Brodeur, Paul. "The magnetic-field menace". MacWorld, July 1990 pp. 136-144.

Cherniss, Cary. Staff burnout. Sage Publications, Beverly Hills, 1980

Deal, Terrence E. , *Pacman meets Godzilla: when technology and people meet,* presentation at 21st Annual Electronic Data Processing Symposium, Falls Creek Falls State Park, April 22, 1991.

Greenberg, J. S. Comprehensive stress management. Wm. C. Brown Publishers, Dubuque, 1987.

Mendenhall, Mark. "What motivates some, may not move others", The Chattanooga Times, April 11 1991.

Pines, A.M., Aronson, E. and Kafry, D. Burnout: from tedium to personal growth. Free Press (Macmillan Publishing Co. Inc.), New York 1981.

Tubesing, N. L. and Tubesing, D. A. "The treatment of choice" in Job stress and burnout, edited by Whiton Stewart Paine, Sage Publications, Beverly Hills, 1982.

Paper 5131
ANATOMY OF RECOVERY
Drawing on Experience

James A. Depp
UP TIME Disaster Recovery, Inc.
Sacramento, CA 95834
916-648-1282

The story is told of a farmer years ago who needed help to work his farm. While in town he interviewed a number of people who wanted the work, most of whom were either not qualified or did not want to make sufficient effort. By the end of the day, the farmer interviewed the last person, asking how the candidate would handle emergencies. The farmer received the unexpected response "I can always sleep through a storm". He received the same response to each specific calamity which he decribed. Frankly concerned about that response, the farmer hired the man anyway, unsure of the meaning of the peculiar response.

Several weeks later a great midwestern storm swept over the area in the middle of the night. The farmer awoke, dressed, and went to the farmhand for help in securing the farm against the rising winds. The farmhand was sound asleep, and would not wake up! Angry, the farmer dashed out to close the barn, climbed to the roof to check the shingles, scurried about the farm making sure everything was in its place and tied down.

To the farmer's surprise, wherever he looked, he found everything in order - already secured - safe from the storm. And he thought back to the farmhand's comment, and realized the meaning of "I can always sleep through a storm."

Preparation for disasters is similar. We have our routine activities, and those activities that protect us when the storm hits. The farmer was unaware of the farmhand's detailed and conscientious work beyond his regular duties - but the farmer knew there were numbers of things around the farm to be secured when the storm came. In our scurry among operations, software development, staff and user training, and planning future applications, various aspects can be overlooked, or deferred. Today we'll review from the eye of the storm - testing and recovery - some of the things which can go wrong, so that you can confirm that procedures in place will protect your operation.

We'll talk first of the people oriented issues, since it is the people - users and systems staff - who are actually

impacted by the disaster. We will then review data communications - access to the computer by the people. Next will come the information and procedures which permit the computer to process - the information. Finally, we'll talk about the hardware configuration. The specific items discussed are drawn from actual experiences either in the test environment, or from the postmortems of actual disasters.

People

It is easy for the systems staff to become isolated with disaster recovery, or to isolate themselves. The two most significant resources are the users and the software support staffs, particularly for purchased software. In many cases neither have seriously confronted the recovery options so some education is required. It is fully worth the time in cooperation, and participation.

It is important to establish who or what is being tested, and to confirm that it is the plan which is being tested and revised, not the individuals involved in the testing. This single consideration will significantly reduce the anxiety of the test situation, and the recovery.

Our clients who have involved the system users in the planning and testing process have had good success. The team spirit has been obvious, and a variety of everyday procedural changes have been developed from the users themselves, avoiding much of the implementation difficulty.

The scheduled full system tests have been well received by the users. The unscheduled fire drill tests must be carefully structured to avoid major disruption, while being of long enough duration that significant data entry or work is not left to redo on the production system.

Specific evaluation is required to determine the best work location following a disaster. Strategies include working individually - telecommuting - from home, working from other company or temporary facilities as clusters of employees, and establishing alternate work space for as much of the staff as possible. The strategy should take into account the informal communication among staffs, and the ease of accountability among staff members. The aftermath of a major fire left many staff members sharing offices in the other company facilities. While the cameraderie of 'camping out' lasted the first week, after that the disruption caused was substantial.

Data communication

The significant enemy in disaster recovery is time, and data communications has been the area of most significant time commitment. If recovery requires network access, establish the alternate access in advance, and test it frequently. When possible, establish redundancy that is routinely used, since spare lines can go down, unnoticed.

In our tests where secondary communications have been used it has taken 2 - 4 weeks longer than expected to install the equipment. In a disaster in Salt Lake City, the choice of relocation was determined by availability of telephone circuits, above all else.

X.25 and LAN technologies provide flexibility if designed with recovery in mind, and if the application is well suited to the communications technology. The move to the network concept should be carefully evaluated, since the specific data movement has in some cases a negative impact on production. The LAN bridging and routing flexibility can be well utilized with careful capacity analysis. Bridges can be positioned to protect against loss of the network, and can then serve very well in disaster recovery.

Procedures

Creation and handling of routine system backup tapes cause a surprising number of obstacles to recovery. If the files are backed up at various times during the day or week, review the strategy in detail, since we have had situations where some files on the system were overlooked, and never dumped, or were dumped so infrequently that they were out of date. This problem caused significant work following one fire, when the source code was not routinely stored, and was lost, causing six months of effort to update a very old copy.

Also review the backup strategy to be certain the output is reviewed to confirm that all files were stored. Database root files have been missing due to single users logged on with a UDC opening the database.

Tape errors do occur, and can be in either the directory or the data files. If the information is important, it is worth verifying the tapes. It is also valuable to review the manuals for store software, particularly compression software, to know the options available if the directory or individual files are corrupted.

The procedures for handling backup tapes should be reviewed, and tightened. One site had a system failure, called for backup tapes from offsite storage, and went into a meeting to discuss the recovery strategy. Failure to isolate those tapes, failure to communicate with the incoming shift operator, and weak rotation procedures resulted in the new operator storing a bad system on the current backup tapes.

Software

Utility and application software runs on one system, so it is likely that they will run on a similar backup system. This assertion is likely to be more accurate for software developed in house, since issues beyond function are built into purchased software. Software licensing and control is becoming increasingly significant, and adds complexity to the recovery activity.

Software licensing and support during recovery should be discussed with software vendors. Approval and support should be expected for backup testing and disaster recovery.

The vendor should describe any procedures to be used in a backup environment. These may be separate run options, such as a RECOVERY mode; or it may be necessary for the vendor to access the recovery backup machine and change program parameters to permit full operation.

The implementation of the HPSUSA number with MPE-XL version 2.2 can further complicate recovery. In general, recovery on the client's operating system is preferred, since any patches and specifics required by the application software are known to be installed. Changing the number in the field is not a difficult activity, but does require an additional step and time in the recovery process. Application and utility software vendors will migrate to use this additional level of license control, so the discussions with the vendor should include their handling of this issue.

System security through passwords, port passwords, etc. should also be reviewed to ensure that documentation and strategy are available in the recovery situation. In one test the application expected a specific device on one port, and failed because it was not available. In another, the software expected DTC ports to be configured as modem or direct and would not permit casual changes.

Finally, if the testing environment requires program use significantly different from everyday performance, attempt to assess the impact and capability of the software. The

specific case was DBRECOV in MPE-XL which in normal use did recover logged transactions to a specific database. But in the recovery test, a program bug would not permit a file to be built large enough to recover transactions for all the databases in one pass. In this case, recovery testing exposed a weakness not likely to be found in normal operations.

Data

Updating the data to the point the disaster occurred is a challenge in most organizations. Database logging provides a vehicle that is not used as often as one might expect, but it is still dependent upon getting a tape out of the midst of a disaster, or of maintaining mirrored data. Where data updates are large, or recovery time short, these are the best options.

Some organizations realized in testing that reentry of data was confused due to filing methods of the source documents, and so changed those procedures to facilitate updating. One organization recognized that following a fire, they might have no paper files left, and that ALL the informantion they would have on the clients would be on the tape. This led to a deliberate monitoring of reference to paper, and addition of certain fields to the databases so that full information could be drawn from the computer.

Testing

Recovery testing was the environment in which many of these potential pitfalls appeared. The testing was valuable to the extent that problems were identified, and overcome, as well as to the extent that the recovery went very smoothly. The pitfall here is that the benefit of testing fail to feed back into future recovery activities.

Testing can require changes in the recovery strategy, and it is our best clients who are continually updating the plan documents, with each test. Batch jobs sometimes require change, and many times the changes can be incorporated into the routine jobs so that future tests and recoveries will not require the changes. Some differences in processors and operating systems might require modest documentation to ease the future testing. In each case, keeping the documents current is very helpful.

Hardware Configuration

Recovery planning and testing lead to a number of critical reviews of the data processing activity. The recovery configuration however obtained will cost some additional funds, and several considerations are appropriate.

The processor itself should be sized for the anticipated backup processing. If the critical processes are fewer than the regular operations, and the users reduced due to housing and data communications, a smaller processor may be possible, and less expensive. Schedule adjustments might also provide options for a period of time.

Terminal and printer availability is both a need and a convenience. If the staff is housed temporarily in clusters or cramped quarters, the recovery system may require fewer ports, fewer communications links.

Critical processes may not require as much disc capacity, but testing is a key part of this decision. Some of the attempts to load only part of the software and data failed, usually due to less than full information on interactions among databases and applications.

Conversely, if applications are very independent, and cyclic in their use, it is possible to rotate applications over time, permitting operation on a smaller system.

Bits and Pieces

Call them loose ends - the things that happen that cause us to simply shake our heads. Testing can be a tense time, and some problems can simply bring a smile. A set of keys forgotten - to the backup tape boxes. A map - to the recovery site.

If action on any of these suggestions provides relief, I've met my objective in presenting them. And perhaps their presentation and resolution will permit, if not sleep through a storm, at least less stress in the midst of it!

Paper No. 5132

Getting the Paperwork Done: Managing a Documentation Project

Pamela Dickerson
ECI Computer, Inc.
1231 E. Dyer Road
Santa Ana, CA 92705
(714) 434-8841

Introduction

When documentation is done well, no one notices. When it's done poorly, everyone notices. Good documentation reduces the number of calls to your response center, which saves you money. Good documentation can serve as a selling tool, enhancing the perceived value of your product. Good documentation educates, informs, reassures and provides tangible evidence of the product, which otherwise consists of a tape, or diskette, or some other magnetic media. Good documentation gives the decision makers within your customer base the confidence that they made the right choice, and encourages them to do business with you again.

Despite this, documentation is rarely considered to be an integral part of software packages. Too often, it is added on at the end of a project, after the coding has been done and, in some cases, after testing has been done. In an ideal world, documentation would be started at the same time the software is started. Since we do not live in an ideal world, however, this paper addresses both the perfect world of software development where specifications are written for everything and planning is an integral part of software, as well as the real world where chaos reigns supreme.

In this paper, you'll learn how to:

- Evaluate what constitutes good documentation
- Determine what documentation is appropriate for your product
- Decide what type of documentation professionals you need
- Select the tools for creating documentation
- Come to terms with schedules, budgets and outside vendors
- Develop a documentation plan

What is good documentation?

Good documentation is documentation that gets used. Good documentation, whether printed or on-line, is easy to reach and easy to use. Good documentation is

written for the user, not the developer. It is organized in a logical fashion and follows industry standards, when they exist.

How do you find good documentation? Take a look around your workspace. If you're like most professionals, you have a number of manuals for hardware and software scattered about. Some of them never get taken off the shelf. Others are marked up, dog-eared and generally unattractive. These are the ones that qualify as good documentation. Look at the manuals and on-line documentation for products that you use and group them into two categories: documentation you use, and documentation you do not use. Then examine the documentation you use to determine why it is effective. Look at the other pile to determine why it is not effective. Chances are, you'll find the roots of good, and poor, documentation in these two piles.

Good documentation is well organized. It is largely self-explanatory in its use. It is not dazzling for dazzle's sake, nor it is boring. It anticipates the users' needs, and meets them. Whether printed or on-line, it is not crowded. Above all, good documentation never forgets the audience which will be using it. Everything about documentation -- its language, style, format, even color -- reflect the taste and attitudes of the audience.

What documentation is appropriate for your product?

There is one theme which recurs throughout this paper, and that is: Know Your Audience. Understanding who will use your product determines everything about your documentation. The type, or types, of documentation that are appropriate for your product is dependent on both the product, and the characteristics of the people who will be using your product.

The only limitation on the number and types of documentation is your imagination. Typically, documentation for software can be broken down into the following categories:

- Data sheet
- Installation guide
- User guide
- Training guide
- Technical reference
- Procedural documentation

Data sheets are product overviews which offer a short (under five page) summary of the product, including features and benefits. These are often prepared by marketing departments, but they provide potential customers with technical information regarding the product. For this reason, you should consider data sheets your first line documentation.

Installation guides instruct users on how to install the software in order to gain maximum benefit. In small packages, this may consist of little more than, "insert the diskette into your A drive and type, 'install.' " In larger systems, where programs must be installed in a given sequence, these documents can become quite large.

User guides are intended to be used when someone is running the program, and they may be printed manuals or on-line help. Quick access and ease of use are key features of these documents. When designing user guides, keep in mind that the user is most likely in the middle of something else when he determines that he needs to look up information.

Training guides are designed to teach the product to a new user. There is a considerable level of detail associated with these guides, which you may assume will be taken home for review and studying. These should be developed in close association with the person responsible for teaching whatever training classes you may offer.

Technical reference is where you provide the information that advanced users need to effectively run and maintain your product. Novice users may never crack this book, and smaller packages may be able to put all their technical notes into a few pages.

Procedural documentation explains how the software product fits into the real-world environment of the user. For example, procedural documentation for a large system would include information regarding where various copies of a report are distributed, and how various people within the organization use the report.

Of course, not every product requires all of these pieces of documentation. If the product in question is a utility for a PC, perhaps a screen saver, or a data compression program, the documentation you need is minimal. You could include a README file on-line, and a small printed manual. Procedural documentation may be reserved for the largest of packages (the type that run entire businesses), and you may dispense with training guides altogether, depending on the type of product you manufacture. The important thing is to ensure that all functions of the documentation are included, even if some are combined.

As you're considering which types of documentation you want to include, you should also determine whether your documentation should take the form of a printed manual, on-line documentation, or a combination of the two. This determination is governed somewhat by cost and hardware configuration, but most products can benefit from some type of on-line documentation, even if it is little more than a README file.

What Goes in Almost Every Manual

Most every printed manual should contain the following components (there are some few exceptions to this rule). Begin by assuming that you need to include the information, then demand good reasons for omitting any of these.

Cover page	Your cover page should include the title of the manual, which should be brief and tell users exactly what is inside. "Inventory Control Procedures" is appropriate; "Procedures for the Intake of Vendor Products by Warehouse Personnel to Ensure Accuracy and Eliminate Errors," is not. Besides the title, your cover page should include the date of publication, your company name, and the document number.
Copyright notice and trademark information	Directly behind the cover page, this lists the year of copyright (be sure to copyright your manuals) and a threatening paragraph concerning what you will do if someone violates the copyright. In addition, this page is where you identify the various trademarks that appear in the text throughout the manual.
Table of contents	This lists the major sections of the manual with the beginning page number of each section. Page numbers are essential. If you use chapters with subheadings, include the subheadings in the contents, as well. Think of the contents as a way of explaining, at a glance, what the user can find in the manual. It is also a yardstick for prospective customers to determine the value of your documentation.
Introduction	Your introduction explains who you expect to be using the manual and the background you assume they have. It also explains how the manual is organized, the way you separate system prompts from user entries, and where to go for more information. This section is not always called an introduction (Hewlett-Packard often calls it a preface, for example), but when your reader is finished with this section, he should be able to use the manual without difficulty.
Chapters	If this seems too obvious, think about the manuals you've seen which begin at page 1 and press onward to page 234 without any breaks. Chapters divide information into chunks which the reader can understand easily, and it also forces you to keep to the subject at hand.
Glossary	Similar to a small dictionary, glossaries contain information unique to your environment. Ideally, this should be a mixture of computer terms and terms used in the industry you are writing for. This small document can ensure that everyone using the manual understands the terminology.
Index	A comprehensive index can be one of the most-used features of your manual. Unlike the table of contents, which lists major sections, the index lists individual items, and is used by readers unsure of where a particular item is located.
Evaluation form	This gives users a chance to help you improve your manual. Designed properly, an evaluation form will not only tell you what is wrong with your manual, but also what is right with it. Information you obtain from this source can be invaluable in designing future manuals.

New user interfaces make it possible to use help windows which the user can access from within a program. This solves the problem of having to exit the program to look up the information on-line. If you choose to use this technology, make sure your developers emulate the industry standards. There is a company which required the user to enter <Ctrl> + F9 to close the help window, a technique which is neither standard nor intuitive. Such ignorance of standard window closing techniques rendered the documentation difficult to use, and resulted in a number of calls to Customer Support asking how to close the window. This defeated the purpose of the documentation.

Selecting documentation professionals

"Anyone can write. We'll just have a programmer who wrote the package write the documentation. She already knows the package, and we will save money."

The above attitude results in more bad documentation than any other single factor. As with all other members of your organization, technical writers are professionals. They write manuals and on-line help for a living. They are well versed in the language, format and style which will make your documentation easy to read. They know the software and hardware which are key to preparing documentation. Aside from everything else, they write considerably faster than non-writing professionals and enable your programmers to concentrate on what they to best: designing and coding product.

Once you've accepted the fact that you need professional documentation specialists, you have two options: in-house, or contract. To determine which solution is right for your environment, consider the following:

- Quantity of work
- Training investment
- Amount of control

When considering **quantity of work**, ask yourself the following question: do you have enough documentation to keep your proposed documentation staff occupied for one year? If you're working on more than one project at a time, or you anticipate project development over several years, an in-house documentation staff will save you time, money, and headaches. Also consider the amount of revisions you expect the project to undergo. If this is a one-time, standalone product, then you should seriously consider an outside consulting firm. If you anticipate updating the product on a regular basis, you may prefer the consistency and prompt turnaround time that an in-house staff would bring.

Your documentation staff, be they in-house or consultants, will require some level of **training investment** on your part. If you plan on using HP hardware and software to produce the documentation, for example, you'll find few professionals already equipped with these skills. On the other hand, if you select a PC-based desktop

The Documentation Team

The following functions may not be assigned as they are outlined in the table. Free-lance writers, for example, fill all these functions. Regardless of how assignments are allocated in your environment, it is important that each function be accounted for.

Manager	The manager is responsible for maintaining the schedule, making sure that the project stays within budget, and running the day-to-day documentation operation. He is part supervisor, part accountant, part software analyst and always a diplomat. The manager is also liaison to the rest of the company, feeding information from the technical side to the documentation staff, and vice versa. He makes the hiring and other staff decisions, and is responsible for the rest of the documentation team. Ideally, the manager should be well versed in each aspect of documentation.
Editor	Technical editors read every piece of documentation that is produced. They are interested in style, content, language and accuracy. They are not particularly interested in punctuation and spelling; that's the proofreader's job. It is up to the editor to make sure that the documentation has one "voice," even if there are 40 writers involved. Editors enforce the documentation standards of the company.
Writer	Writers, or documentation specialists, actually produce the documentation. They research the product to determine how it works, and then communicate that to the users. Writers interview technical staff, make revisions to their own and other writers' drafts, and almost always work under deadlines. Depending on the environment, writers may only work on one product, or may work on many different types of documentation.
Proofreader	Proofreaders read all the documentation for spelling, grammar and punctuation. Although there are software packages which mimic some of a proofreader's duties, the human proofreader is not yet obsolete.
Production staff	Production staff prepare the documentation for reprographics and distribution. This can include pulling tabs to be inserted, working with the shipping department to send documentation to customers, and working with outside printers and binderies. These team members are not so concerned with content as they are with appearance.

publishing solution with laser printers, you'll find the number of prospective employees/consultants increases significantly. (As an aside, you, not your consultant, should determine the tools you use to produce your documentation; this will be addressed more fully later.)

When it comes to the **level of control** you want to exercise over your documentation, that is a decision which only you can make. If you are seriously considering outside consultants, investigate them as thoroughly as you would a prospective applicant. Talk to other clients regarding billings, deadlines, accuracy and quality of work. Determine, as best you can, if the company will be in business in five years. If you're considering a one- or two-person team, attempt to discern not only if they will be in business, but where they will be located. It does you no good if they have moved halfway across the country the next time you need them. Ask if they can assure you that the same people work on all your projects, or if they can assign a single project manager to your account.

Regardless of how many documentation people you eventually have on your team, one person has to have overall responsibility. If you're working with an outside documentation group, you'll be less involved in the day-to-day management, but the following warnings about managing the documentation project still apply.

Make sure that the schedule is updated regularly and that it is understood by all members of the documentation team. Let them know when due dates are approaching and what will happen if the due dates are not met. This does not mean, necessarily, that you should shoot the junior writer if the project is not completed on time; it recognizes instead that most writers are notorious procrastinators, and many work best under threat of impending disaster. Instead of setting one deadline for six months down the line, set intermediate deadlines so that you have control over where the project is at a given point in time.

Provide as much reference material as you can. In the ideal world, everything would be coded from specifications, and the specifications would provide the base material for the documentation. In the real world, writers will appreciate anything that provides information, including notes on napkins from impromptu lunch meetings. Also work on developing a style guide for your writers. This includes information on how your company writes its manuals. For example, if your company uses the term, "enter" instead of "input," that should be noted in the style guide. Such a manual will ensure that your documentation has the same "voice," no matter who writes it.

Monitor the documentation staff to make sure that each person is performing his function completely, and that his function is also his primary task. There is nothing wrong with an editor writing if there's a crunch, but it's not his primary function. The more everyone operates within their realm of expertise, the smoother the project will flow.

Hold regular, brief meetings to make sure the project is on schedule. This is where you gain the information necessary to keep the schedule updated. This is also where

everyone on the team keeps informed as to where delays are likely to occur. If you have to schedule a 70-hour workweek to overcome a delay, your team will be much more likely to accept it if they have seen the reasons behind the delay in the first place.

Selecting the Tools

There comes a point when you have to decide which tools are most appropriate for developing your documentation. PC-based tools provide the greatest flexibility, but if you will have to dedicate a number of PCs to the documentation effort. Unless your documentation requires a number of graphics in the text, you'll find that the more powerful PC-based word processors are sufficient to meet your needs. Two of the most powerful are WordPerfect and Microsoft Word. If you want the increased flexibility of graphics (you can insert graphics with both products mentioned above), investigate industry leaders such as Ventura Publisher and Aldus PageMaker.

If you stay with industry leaders on the software side, you'll find it much easier to find documentation professionals. There are a number of professionals who are familiar with a wide range of desktop publishing software, and someone who knows Word can learn WordPerfect quickly, and vice versa.

Unfortunately, there are often a number of factors outside your control which determine the tools you are able to use. Chief among these are financial constraints, corporate strategy and company alliances. If you are operating within a tight budget, investing in new equipment may not be an option. You'll need to find software that works with your current hardware configuration. If your company is dedicated to going to a PC LAN in the near future, your documentation department is likely to be a part of that plan. Compatibility then becomes a key issue in the hardware, and software, you install. At the same time, if you are an HP VAR, you may enjoy certain advantages by using HP hardware rather than bringing in, for example, Macintoshes.

Besides tools specifically designed for printed documentation, you will also need to consider tools for on-line documentation, and, if your product runs on anything but PCs, tools for accessing the product itself. Terminal emulators, screen capture programs and on-line documentation builders are all products you should investigate at this stage. If you have access to something similar to NewWave, make sure that you take full advantage of it.

Make sure that you have the right printer for your documentation department as well. You can save a considerable amount in typesetting charges if you invest in a laser printer, but make sure that the printer you select is compatible with the software you are running. Also, be sure that your printer has enough memory to handle whatever graphics you intend to include in your documentation.

Besides a printer, you'll need a variety of fonts, as well. Some printers have print cartridge slots where you can install font cartridges. This eliminates downloading

soft fonts and is a good place to put workhorse fonts that you use often. The LaserJet III printers come with a number of internal fonts which can be scaled to the appropriate size, thus changing the need for fonts. Carefully investigate which fonts are included with your printers and which you will need to obtain from other sources.

Consider your documentation tools for what they are: capital investments. Invest in the right tools, and your documentation department will be able to produce high quality documentation at a high level of productivity. While the initial hardware and software cost may be more than you initially considered, the long-term benefits are great.

Schedules, budgets and outside vendors

When you begin the documentation project, invest in a good project management system. This enables you to track progress on the documentation without spending an inordinate amount of time doing so. Packages which will automatically produce Gantt charts and PERT charts are ideal, so that you can identify where problems are occurring.

The best project management software only works as well as the information that is put into it, so take your time when you develop your schedule. To begin putting together a schedule, start at the end. In an ideal world, documentation and software are released together, and that should be your goal. Look at the software release date, and make that your *documentation release date.*

Now determine how your documentation is getting to the customer. It is best if you can include it with the software, but some vendors ship it separately. Regardless of whether you ship your documentation with your software, or send it separately, determine how the shipment will take place (by what carrier), and speak with them to determine how long the shipping will take. Subtract that many days from the release date, and you have your *shipping date.*

Decide how you are going to reproduce your documentation. Offset printing is best for readability, but it is expensive. Talk to your printer (even if it is in-house) to determine how long it will take for him to produce the manuals once you give him the originals. While you're speaking to the printer, find out if they pick up and deliver, or if you'll have to make special arrangements. Will they ship directly to your customers (and do you want them to do so)? Take the number of days your printer says he needs, add a few for safety (printers are notoriously unreliable) and subtract this from your shipping date. This gives you the date your documentation originals have to be at the printer, or the *"drop off"* date. If you're having your originals typeset, include that time here, as well.

Let us examine a project with a software release date of November 1, 1991. Shipping takes one week, which means that your product has to be out the door October 25. Your printer tells you that your job will require 10 working days, and

you give him three extra days. This means that your "drop off" date is October 8. So at midnight on October 8, your documentation should be ready to go to the printer. This does not mean that you can be making changes to the documentation until midnight on October 8, however.

Most documentation requires preparation before you send it to the printer. You'll need to estimate the number of pages in order to estimate the printing cost. You need to bundle everything up carefully and label it as to the number of copies to be reproduced. Is the printer inserting tabs for you? If so, you'll need to indicate where the various tabs go within the documents. And, you'll need to make sure that the printer either has the tabs in stock, or can obtain them by the drop off date. If you're including any other special materials, such as a reply card, you'll have to arrange for those to be sent to the printer, as well. If you allow a week to prepare the shipment (including gathering artwork, indicating tabs, and so on), the last day that you can make changes to the documentation is October 1.

From here, scheduling takes on a more nebulous aspect as it is tied closely to the software. A good rule of thumb is to use the first day of testing as the last day of major changes to the documentation. This enables the documentation to be tested with the software, and also frees your documentation staff to concentrate on making changes rather than creating whole new sections. Changes to the documentation from this point forward should be as a result of testing. It's too late to incorporate philosophical changes in the doc, now. Think of this as the *test date*. In our example, let's assign a test date of September 1.

This means that you have until September 1 to prepare drafts, have them reviewed, and correct them before the documentation goes through the official test routine with the product. Unlike programs, which are rarely examined on a line-by-line basis, documentation has to be. Documentation must be reviewed for concept as well as content. After the first draft of a section is written, it needs to be reviewed by a member of the technical staff for accuracy.

Typically, documentation goes through two technical reviews: the first is a "meat" review where the reviewer checks for the accuracy of the detailed information. The second is a "gravy" review where the reviewer is checking the conceptual premises. These reviews can be combined in small organizations or on small projects, but should be handled by two different people in larger organizations or on large projects.

Do not confuse technical reviews with editing. After all technical changes are made to a document, it needs to be edited by someone other than the person who wrote it. Large companies may have a designated editor; smaller companies may have another writer look over the document. Editing looks for style consistency, language, the flow of the document as well as proofreading tasks such as spelling and punctuation.

At this point, we know that we have a test date of September 1, and we know that our documentation is going to be reviewed and revised between now and then. So,

beginning with the project start date, work with the technical staff to determine in what order you are going to approach the project. Read any material you can find. Your goal is to have an idea of how long it will take to document the project (how long to produce a first draft), and how long reviews and revisions can be expected to take.

When it comes to **budgets** and documentation, there is never enough time or money to do the job that most documentation professionals would like to. Recognize that you operate in an environment of limited resources, and you will end up with a high-quality product. Invest money in the areas where you will see the greatest return: people and equipment. You can hire a writer fresh out of college, but you will spend time training them in areas that an experienced writer already knows. If you have the time, the savings associated with hiring an inexperienced writer can be significant. If you do not have the time, it's not a wise place to spend money.

There is a similar trade-off associated with equipment. If your printed documentation is going to include pictures of the screen (and it should), the cost associated with a screen capture program should be weighed against the time it will take your writers to manually type the screen into the documentation.

In short, the rule here, as with any budgeting area, is: Live within your means.

All but the largest companies end up working with **outside vendors** at some point in the documentation process. Most companies use outside printers to prepare manuals. Some use typesetters, and many use companies which manufacture custom binders. You may even use an outside vendor to write your documentation. Be aware that when you use these outside vendors, you lose some amount of control. Keep in touch with them to ensure that the project is proceeding on time and within budget. If the end result is not of the quality you were expecting, or it was later than acceptable, let the vendor know, and consider another source for your future needs. Likewise, if the vendor exceeds your expectations, let him know that, too. Such feedback will improve your relationship with them on future jobs.

Putting together a documentation plan

Begin your documentation plan by determining the various types of documentation you are going to need, and what the purpose of each component will be. Write up a brief description of each component. Doing this makes writing easier, helps reviewers, and aids you in keeping control of the project. It's easy to spot when the project begins to get off track, and you can bring it back in line without much difficulty. If you decide as the project progresses that your preliminary ideas are not working out, you have something concrete which you can change.

Next, develop a user profile. For example, are you writing a package for other computer professionals, or will your users be computer neophytes? The language, style, format and content will vary depending on who you expect to use the documentation. Such a profile might read like this:

> The user of the ABC system possesses little or no background with computers. He is most likely a high school graduate with no further education. His main reason for using the computer will be to determine the location of the auto parts within the storeroom in order that he may ready them for mechanics. He will also be responsible for checking that the correct parts arrive from vendors and entering that information into the system. It is important that he be able to find the information he needs with a minimal amount of effort.

Your documentation plan should also include sample page layouts. These don't have to be fancy, just penciled outlines of what part of the page will have what type of information. Each page should have a page number, and you may want to include a version number and revision date, as well. Other considerations, such as type styles, should be included here as well. Also include how the documentation will be presented. Will it be permanently bound, like a book? Or will you use three-ring binders, which make updates easy? Will you be updating your product and sending changes to current users? If so, you'll need to make provisions for updating your documentation, as well. The format section of your documentation plan may read something like this:

> The documentation will be in two columns with the action to be performed on the left and the instructions for doing so on the right. It will be bound in looseleaf notebooks which will allow updates to chapters to be inserted later on. Notebooks will be 8 1/2" x 11" and will be sent with the initial order of documentation. Preprinted tabs will separate the chapters. There will be a table of contents, copyright notice with trademark information, index, glossary of terms and an evaluation sheet in each manual. Updates to the documentation will be included with software updates.

Your documentation plan should also include profiles of the types of people you will need, the number of people you need, and a budget for the project's life. Finally, your schedule will make up the last part of your documentation plan.

A well-thought out and executed plan will make the documentation process run smoothly, and will ensure that you get the best documentation you can.

Summary

Documentation is part of a product that, when done properly, attracts very little attention. Yet its contribution to the overall success of a product is significant. High-quality documentation reduces the number of calls to your response center, and increases the level of customer satisfaction. It provides sales information while keeping users informed as to how to use the system. With careful planning and a corporate dedication to high quality products, a documentation project can be managed to provide good documentation on time and within budget.

The World's Smallest Style Guide

The following information provides simple guidelines to keep in mind as you prepare your documentation.

Two columns	Use two columns for straight text. Put the subject in the left column and the detail in the right. This format allows the reader to quickly scan the text until he finds the section he needs.
Vertical lists	Number your lists only if precedence is important. For instance, if it makes no difference what order the items have, do not use numbers. If, however, one item must be done before another, number them.
Use illustrations	This is particularly true when explaining screens. Include a picture of the screen, or an illustration.
Use fonts and colors sparingly	It is difficult to imagine when more than two colors would be appropriate for a single page of instructions. The problem with fonts and colors is that they are fun to work with, and can be overused to distraction. Consistency is especially important with fonts. Generally speaking, one or two fonts with their associated light face, bold and italics works well. If your first reaction when you look at a page is, "Wow! Look at all the fonts!" there are probably too many. Use boldface for title or subheadings; use light face (regular) in capitals for other important areas. Avoid putting boldface in all capitals as it is very difficult to read. In fact, avoid using boldface for entire paragraphs or even sentences. If something is of extreme importance, call attention to it with a NOTE, CAUTION, or WARNING notice. All capitals are difficult to read since most of us read by word shape, and words in all caps are all rectangles. Since you are trying to make things easier on your reader, help him out here. There is also the issue of serif vs. non-serif fonts. Serifs are the little lines attached to letters such as "s" and "t." What you are reading now is serif. Sans-serif fonts have a modern look to them, but they are harder on the eyes. Use sans-serif for headers and serif fonts for text.
Layout	Use similar layouts for similar information. If you have to describe screens in four different sections of the text, use the same page layout for each section. Use plenty of white space, and keep the layout simple.
Second person, active voice, present tense	Without going into a grammar lesson, this means use the subject "you" in your writing. Instead of, "The inventory control number is entered by the user," try, "Enter the inventory control number." Avoid mixing present and future tense; stick to the present tense. Thus, "Enter the number and the system displays the item's location." Not, "Enter the number and the system will display the item's location."
Jargon	Use jargon which is appropriate to the industry. This is especially important for tutorials or examples as it enhances the documentation's credibility.

Service Level Agreements - Only As Good As The Data

Brian Duncombe
Strategic Software Group Ltd
4246 Dunvegan Road
Burlington, Ontario
Canada L7L 1P8

Phone
416-632-2773

Fax
416-632-8704

!uunet!ssgltd!brian
brian@ssgltd.COM

ABSTRACT

In this presentation, the author will give an in-depth description of Service Level Agreements, what they are and what they should be including the following:

- Just what is a Service Level Agreement (SLA)
- What a Service Level Agreement should Include
- How do you develop and implement a strategy.
- What data do you need before formulating an SLA.
- Who is responsible for establishing and monitoring.
- How do you insure that it is working.
- What happens when the agreement is breached.
- Do you really need an SLA (And Can You Afford It)
- Alternatives to the traditional SLA.
- Advantages of an SLA

WHAT IS A SERVICE LEVEL AGREEMENT (SLA)

A Service Level Agreement is a contract between the Information Systems provider and the Information Systems Consumer. It defines the expectations and obligations of each party as well as the methods for resolving conflicts.

The following is an example of a basic SLA :

SERVICE LEVEL AGREEMENT

THIS AGREEMENT dated as of August 6, 1991 is entered into by and between

The Accounts Payable Department, a functional unit of Acme Screw Enterprises Inc. (hereinafter called "AP")

AND

The Management Information Systems Department, a functional unit of Acme Screw Enterprises Inc. (hereinafter called "MIS")

WITNESSETH that in consideration of the mutual covenants contained herein, the parties agree as follows:

1. EXPECTATIONS

The party of the first part ("AR") agrees to limit their demands on and use of the services to a reasonable level.

The party of the second part ("MIS") agrees to provide computer services at an acceptable level.

2. PENALTIES

If either party to this contract breaches the aforementioned EXPECTATIONS, the breaching party must buy lunch.

IN WITNESS WHEREOF the parties have executed this agreement as of the day and year first above written.

By: _____ _____

Title: _____ _____

Witness: _____

Date: _____

WHAT SHOULD A SERVICE LEVEL AGREEMENT INCLUDE

Like any effective contract, a Service Level Agreement should specify in a clear and unambiguous way what the parties to the agreement understand to be the specifics of the covenant. As a matter of practicality, it cannot cover every eventuality but it should spell out in clear and precise terms what is understood to be the "guiding principles" of the agreement. The wording should not leave to interpretation any of the key points of the agreement. Of course, in order to be workable on a daily basis, it must be negotiated to the point where both parties feel comfortable with the terms and where few if any conflicts are likely to arise. It should address the following subject areas:

THE PARTIES INVOLVED

It is imperative that there be no question as to who is covered by the agreement. It should be made very clear which divisions, departments, work groups and individuals are to be subject to the conditions of the contract. If there are to be any exceptions, they should be specifically identified and their unique requirements and/or responsibilities provided for as part of the agreement.

SCOPE OF THE AGREEMENT

What are the bounds of the agreement? Do we limit the responsibilities to the normal working day or must we include evenings, nights, weekends and statutory holidays. What are peoples expectations in the unlikely event that some natural disaster strikes. What happens to the agreement if a major change in the operation of the business such as a new division, department, product line or an unanticipated change in the business volume takes place.

DEFINITION OF THE TERMS USED IN THE AGREEMENT

It is not adequate to use terms such as "acceptable" when describing the level of the service or even some aspect of it. What one person thinks is acceptable under one set of conditions is not necessarily the same as another person (or even the same person) thinks under a different set of circumstances. In order to work on a daily basis, the definition of specific metrics and measured levels must be agreed upon.

EXPECTATIONS OF THE PARTIES

The consumer of the service generally expects that the service will be available in adequate amounts during the periods of time when he/she needs them. There is also an implied expectation that in the event of a conflict, their work should be given more priority than the other consumers of the service.

The supplier of the service assumes that the workload follows some pattern that can be defined and planned for. It is this predictability of user requirements that service providers look for. When the demand for a service varies dramatically, the supplier of that service has a very difficult time managing the resources involved.

HOW THE SERVICE LEVEL WILL BE MEASURED

There is an old maxim that if it can't be measured then it doesn't exist. In point of fact, for SLA's, if it cannot be measured and quantified then it is going to be very difficult to determine whether or not the parties are living up to the agreement.

In drafting a service level agreement, you must be very careful to include measurement specifications for only those metrics that you are actually capable or willing to gather and report on.

HOW THE SERVICE LEVEL WILL BE MONITORED AND REPORTED ON

The agreement must specify what the metrics are as well as the acceptable levels for each metric. Once the items of information required to determine the criteria as specified in the agreement have been determined, the required data collection and reporting must be defined.

DURATION OF THE AGREEMENT

Of course, we all expect that the relationship between the two parties will continue forever, just as we expect the enterprise to continue. What is meant by the duration of the agreement in this case is a point in time at which the agreement will be automatically re-negotiated.

METHOD OF RESOLVING DISPUTES

In spite of the best of intentions, at some time the conditions of the agreement will be breached by one or both parties. The agreement should contain at least the "general principles" of the procedure to be followed and the actions to be taken in such an eventuality.

HOW THE CONTRACT WILL BE TERMINATED

It has been my experience that when a member of the legal profession reviews a contract that his client is being asked to sign, the first thing that the lawyer will do is examine how the contract can be terminated. It is also my experience that the lawyer who drafts an agreement takes special care to ensure that any termination of the agreement will only be allowed to occur if it suits the best interests of his client.

HOW DO YOU DEVELOP AND IMPLEMENT A STRATEGY

In order for you to have even a hope of being successful with an SLA, you must ensure that there is solid commitment from higher levels of management to it's success. If the idea is sold at the highest levels within the organization, many smaller problems will get ironed out quickly and an effective arbitrator for more serious disagreements will already be in place.

At the outset, it should be decided whether the SLA being developed is to be a "hard and fast contract" or a formal mechanism for identifying problems as they begin to develop. If the objective is an "adversarial" contract, then the contents of the agreement should be studied very carefully to make sure that they are as fair to both parties as possible. If, on the other hand, the objective is to define the mechanism for an "early warning system" then the conditions can be much more restrictive. As the conditions of the agreement become harder to keep, the number of incidents of one or both parties breaking the agreement will rise. This will make the agreement and reporting very sensitive to what is actually occurring and provide a delicate barometer of the service. As the conditions become less restrictive, the value of the agreement as an early warning indicator diminishes but the agreement requires less time for intervention and problem resolution.

It is always a good idea to start out with a successful pilot project. To ensure that this project has the highest possibility of success, it is best to chose a user community with whom you already have a good working relationship. If this user does not represent the major consumer of your services then the agreement will be much easier to live up to. By choosing a non threatening pilot, you gain the experience of developing and implementing an SLA without the added aggravation of a hostile consumer. Once the pilot project has been put in place and some experience has been gained with it, you will be in a much better position to customize and negotiate the process for the more demanding client bases.

WHAT DATA DO YOU NEED BEFORE FORMULATING AN SLA

In order to make an SLA work on a practical level, the specifics of the criteria must be reasonable in the eyes of both parties to the agreement. If the agreement promises unrealistically good service then the provider will always be viewed as a under-achiever. If the specifications for the agreement are too forgiving, the consumer will be unhappy and the agreement will again not be workable.

Before you can specify the details of the service levels to be met, you must have a solid understanding of the current and historical service being delivered. This information allows you to set practical levels for the services covered by the agreement. The following give a checklist of the types of information that should be available during the negotiation phase of developing the agreement:

SYSTEM AVAILABILITY - How much of the time has the system actually been available for use? At the very least, this should be broken out by prime shift, evenings and nights as well as weekends and holidays. For periods where the system was not available, the reason should be determined and noted. If you have kept an accurate and up to date system log, this will be of great assistance in providing the basic availability patterns. If it is possible to determine, it would be useful to know when and under what circumstances the system was operational even though one or more applications were unavailable. These occurrences could be caused by scheduled or unscheduled database re-organization, unusual processing or data recovery activities.

RESOURCE CONSUMPTION - Where possible, some historical data should be gathered to show what demands the various workloads place on the primary resources of the computer system. A rough indication of CPU and disk resource consumption can be found from the basic accounting services provided within the operating system.

RESPONSE TIME DISTRIBUTION - This information will not likely be easily determined unless you have been collecting this type of data as part of the use of one of the available system monitoring or trend analysis packages available. If it is available, it should be broken down as much as possible so that logical groupings of users and applications can be characterized. Ideally, this information should include transaction counts that correlate with the response times.

USAGE PATTERNS - It is not enough to know how much of the various resources a particular activity will consume. This information must be combined with the timeframe in which this consumption will occur. In addition, it is important to know how much flexibility there is for minor changes to these patterns. Any usage that can be adjusted so that it will take place at some other time gives the supplier of a service a valuable tool for distributing workload into periods of lower activity. Of course, the consumer must be aware of how certain tasks will be viewed in this light.

BATCH TURNAROUND STATISTICS - Before entering into any discussions about service levels, it is imperative that we know something about the nature of the key batch processing tasks within the organization. This would include information such as actual run times as well as any scheduling requirements and conflicts between jobs.

WHO IS RESPONSIBLE FOR ESTABLISHING AND MONITORING

Both parties to the agreement must take an active part in both the establishment and the monitoring of the Service Level Agreement. If one party attempts to coerce the other party, then the terms of the agreement are likely to be one sided. If only one of the parties is monitoring the agreement on an on-going basis then many of the advantages of the SLA will be lost. In most cases, there should be a very equal participation by both parties in the development and refinement of the agreement. As a matter of practicality, it is usually the supplier of the service that has the most involvement in the monitoring activity simply because the automation of most of the data collection and reporting falls under the control of the service provider. The review of the reports should be an activity that both parties participate in equally. In order to ensure this participation, it is a good idea to consider holding a regular meeting for the purpose of reviewing the reports. This meeting need not be a formal event, in fact it is probably more advantageous to make it as informal as possible, just so long as it takes place on a regular basis.

HOW DO YOU INSURE THAT IT IS WORKING

It is imperative that the parameters of the agreement be monitored on a regular basis. When a disagreement or breach relating to the agreement occurs, having a history of the patterns of the metrics can contribute a great deal to the successful resolution of the problem. By providing regular reporting that allows both parties to track their adherence to the agreement, a developing problem can be detected in it's early stages and resolved before it becomes a major concern.

A procedure for collecting the necessary raw data and conveniently reporting information derived from that data must be agreed upon. The key to long term success seems to hinge on how accurately a few key indicators are identified, specified and appropriate levels specified for. By including a vast array of metrics, the data collection and reporting becomes onerous and the people who are responsible for reviewing the reports will be overwhelmed by shear volume. By reducing the criteria to a few "key indicators", the monitoring, reporting and review will be much more workable on an on-going basis. If the procedure is not automated, it will gradually fall into disuse and the whole agreement process will lose much of its value. The required metrics must be presented in a form that facilitates painless review and that makes it easy to spot problems as they develop. Graphical presentation of the information is probably the most convenient format for the reporting purposes. All parties concerned can easily see how the actual values of the metrics compare to the specified limits. Additionally, graphical formats permit easier detection of developing trends. As has often been stated, a few well organized charts or graphs are worth boxes of printed tabular reports.

Overall system availability is a universal metric. The definition of availability is not so universal. It should be made clear when the service is expected to be available as well

as the times when scheduled down time will have the least impact. If one party makes an assumption about such things as the availability of the service during evenings, nights, weekends or holidays without informing the other party of the expectation then conflict is more likely to occur sooner than later.

For interactive applications, the level of service as seen by the consumer is most easily defined by response time. The supplier of the service should be more interested in the demands placed on his service by the customer. The system cannot be expected to perform within the agreed upon limits if the workload changes radically in either volume or timing. These demands, in the form of user counts, programs run, transaction rates and consumption levels of the major computer system resources should be the key points of interactive service levels. If the agreement says that a certain application will not exceed a fixed number of transactions per minute, then the service supplier has just cause to raise the alarm if the pattern exceeds the specified rate.

For batch oriented workloads, the turnaround time or the completion time is a good place to start defining the level of service. Of course, the agreement should also spell out the frequency of the major processing tasks as well as the expectations of how re-runs will be accommodated.

Figure 1 shows a graph combining response times and transaction rates for interactive processing by a selected group of users for a single working day. It is quite easy to take a quick look at this chart and visualize the characteristics of the workload.

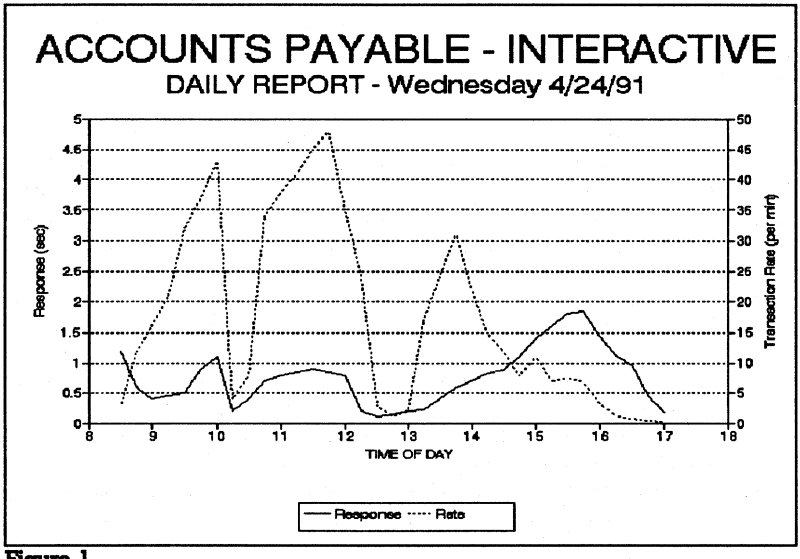


Figure 1

Figure 2 shows information for the same interactive group of users but showing transaction volumes for a period of several weeks. This chart provides a simple method of showing evolving trends. In addition, by setting the limits for the chart to those specified in the SLA, we can easily see how close the actual workload approaches the allowable limits. In the case of the figure shown, we can see that for a period of several days within the period, the limits of the SLA were exceeded.

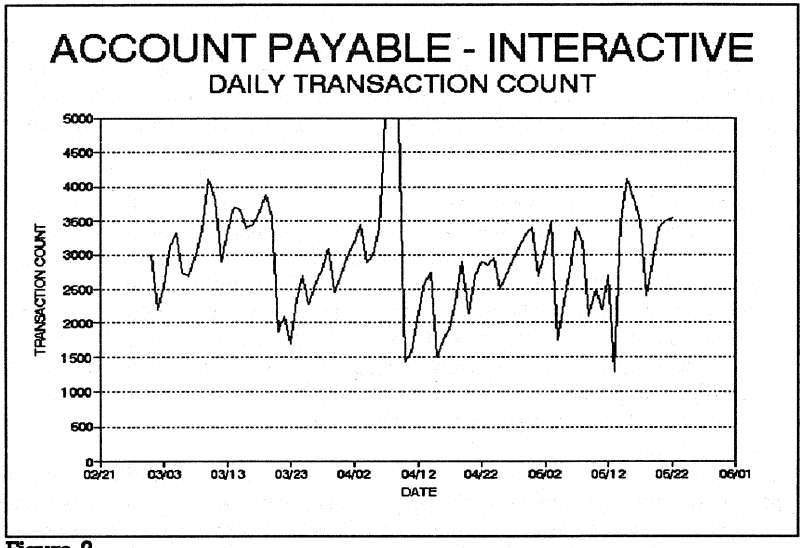


Figure 2

WHAT HAPPENS WHEN THE AGREEMENT IS BREACHED

This should, of course, be specified in the SLA itself. While the agreement specifies what should happen in theory, what often happens in practice is something else. Some metric of the service level could exceed the specified limits on an occasional basis. Most often, this will occur on a very exceptional basis and the parties to the agreement must realize that this is not a particular cause for alarm. The purpose of the SLA is defeated if the limits for the metrics are loosened to the point where they are never exceeded.

When some condition of the agreement is broken, the parties to the agreement should meet informally to discuss the cause of the occurrence. If it was the result of some unusual event that is not likely to occur again then the incident should be ignored. If a recurring pattern appears to be emerging then the breach should be treated more seriously.

After getting down to the root of the problem, a decision will have to be made as to whether the conditions of the SLA are in need of adjustment. If adjustment is needed, then this should be done as quickly as possible. If, on the other hand, the conditions of the SLA are considered by both parties to be valid then the cause of the problem must be determined and alternatives for curing it must be developed in a timely manner. Most likely, one or more short term solutions can be identified and put into place as a stop-gap measure. Identifying longer term solutions may be more difficult because they will likely involve changes that require budgetary resources which are most often difficult to acquire in a short time period.

Like all performance related problems, those arising from breaches of the SLA can be approached from two directions. Simply put, either the demand must be reduced or the supply must be increased (or some of each). Where possible, actions can be taken to either reduce the total demand for the service or re-distribute that demand into periods of time when the supply is more plentiful. Cases where the problem arises because all the activity occurs in the last two or three days of a month can often be corrected by beginning the processing a day earlier or continuing it for a day longer. If the demand is a peak during certain periods of the day, perhaps re-scheduling coffee breaks, lunch breaks or starting and ending times of the workday for the people involved may prove effective.

DO YOU REALLY NEED AN SLA (AND CAN YOU AFFORD IT)

Many of us look upon such formalized agreements as applicable only to larger organizations. We pride ourselves on our ability to "manage by the seat of our pants". The prevalent feeling is that we have always been able to deal with a situation as rational people at the time that the situation occurred. A common belief is that the exercise of developing an SLA would open up subject areas that are better left alone

and create issues that were not there in the first place.

Most people involved in the delivery of Information Processing services will admit that historically their normal mode of operation has been much closer to "reactive" or "crisis mode" than "proactive" or "preventative" management. The emerging industry trend is to treat information resources just like any other asset of the company. This means that we are likely to be much more frequently removed from the cloud of mystique that we have successfully hidden in and forced to justify our use of corporate assets much more honestly and using the conventional metrics of results.

The bigger question is, can you afford not to have Service Level Agreements with you major user bases.

ALTERNATIVES TO THE TRADITIONAL SLA

My feeling is that there really are no alternatives. In order to provide services to our users and avoid the inefficient fire fighting activities of the past, we are going to be forced into a more formal management approach. As computers lose their aura of magic, we will be called upon to manage the computing resources of the company in the same manner as any other capital asset.

Having said that I see no alternative, let me say that I don't think that a signed, sealed and notarized document is the answer to most situations. My own initial reaction to the "simple" agreement presented near the beginning of this paper was one of revulsion. It seems to me that this is actually the last resort, the step to take for your own protection when all else fails. Once you have actually drawn up a formal document, it seems to me that you will have polarized the parties into two armed camps and the battles will already have been fought and the war lost. A formal contract that attempts to resolve day to day issues is doomed to failure. If you are forced to "get it in writing" then the objective, in my opinion, is not worth the aggravation.

The lesson to be learned from Service Level Agreements is that we must communicate and cooperate if we are going to be successful. As long as both parties represent factions within the same corporate entity, the ultimate goal is corporate success. By breaking down the barriers of conflict within the organization, we are likely to have a positive effect on the overall picture. In order to be successful within the organization in the long term, we must develop channels of communication both within our own departments as well as across departmental lines. The exercises involved in the various tasks of developing an SLA are the end in itself. Company politics is not likely to go away and so we will continue to compete within the organization but we must focus on the positive aspects of competition. One of these positive aspects is the reward that comes from nurturing job satisfaction and a pride in our work as embodied in such concepts as the highly successful "Quality Circles".

Service Level Agreements will become a necessity in the future. Not the written contract that we can wield against our adversaries but rather the steps that we can go through in order to develop a team spirit and focus that builds upon a foundation of understanding and cooperation. The education of our user community to the problems that we face as service suppliers as well as our education in the concerns of our users will be a vital component of our success in the future.

The key to success in the future, as in the past, will be good service. In the emerging competitive environment and tighter economical conditions, the concept of good service will truly emerge as the factor that differentiates one competitor from another. Not our own idea of good service but that of our customer's. This teamwork must begin within our own organizations.

ADVANTAGES OF SERVICE LEVEL AGREEMENTS

FORUM FOR DISCUSSION - The activities involved in the formulation of an SLA provide a beneficial forum for the discussion of the concerns of both parties. In the process of negotiating the agreement, it is likely that both parties will spend more time with one another than would otherwise be the case. By discussing the mutual concerns and objectives, the two groups will undoubtedly be more likely to form an alliance for the future rather than becoming adversaries. Formal and informal avenues for future communication and cooperation should be a natural outgrowth of the SLA experience.

SEE OTHER PERSPECTIVE - Of necessity, the two parties will be required to present their view of the requirements that must be embodied in the final SLA. As these requirements are put forward and refined through discussion and negotiation, both parties will develop a better understanding of the problems and challenges of the other. As each gets a better insight into the concerns of the other, many of the minor concerns should be able to be ironed out and the larger concerns addressed in a united effort.

INPUT TO BUDGETING PROCESS - The details contained within the SLA provide an excellent basis for discussion of budgetary requirements. A well developed agreement shows upper management that the parties are taking a proactive approach to the management of the assets within their departments. Reporting from this process will yield valuable patterns and trends that can be used as a basis for future budget allocation as well as backup data for the re-distribution of existing funding.

EARLY WARNING OF CHANGES - Perhaps the most important single benefit of the SLA process is to put into place a mechanism for timely and accurate feedback of information reflecting the service levels within the organization. A well designed monitoring and reporting system will allow changes in business patterns and emerging potential problems to be identified and dealt with before they become critical.

HOW TO TELECOMMUTE AND RETAIN YOUR SANITY

PAUL EDWARDS
COMPUTER RESOURCE GROUP, INC.
1506 ESTATES WAY
CARROLLTON TX 75006
(214) 242-6660

HOW TO TELECOMMUTE AND RETAIN YOUR SANITY

ABSTRACT

Telecommuting is becoming a very popular form of employment. There are many factors to be considered to make this new working environment successful. If the telecommuter is properly prepared, then this can be a very efficient situation for the employee and employer. The telecommuter could work part-time or full-time. Extensive requirements definition are a must.

Proper equipment, software, office environment, work attitude, and comprehensive employer policies and procedures are vital for success.

An in depth discussion of these areas will be provided and practical examples of personal experiences will be presented.

HOW TO TELECOMMUTE AND RETAIN YOUR SANITY

Telecommuting is a fast growing form of employment. It can be used effectively by full-time employees of companies or by part-time entrepreneurs. We will discuss the particulars of this work environment in depth. The amount of success enjoyed by the employee is directly related to good planning with realistic goals in mind.

This presentation covers the definition of telecommuting, justification, equipment, software, office environment, work attitude, and employer policies. Most of the material presented is based on many years of experience in this field.

DEFINITION: The word telecommuter is made up of two words, TELE and COMMUTER. Webster defines TELE as far off, distant, remote/at, over, from, or to a distance. And COMMUTER as a person who travels regularly between two points at some distance.

The corporate employee may be telecommuting on a part-time or full-time basis. The determination of type is based on the justification needed.

JUSTIFICATION: Sometimes the talent required for specific tasks is not available locally. This talent could be made available to a company by a telecommuter. A personal situation of an employee, such as pregnancy leave, illness in the family, or handicaps or other special cases of valuable employees can provide the justification needed. The transportation costs or restrictions of certain areas of the country may make telecommuting necessary.

EQUIPMENT: Office machines, computer systems, and furniture must be purchased or provided to properly set up the office environment.

Basic office machine inventory should consist of a telephone, fax machine, and copier. The neighborhood printer or office supply store might provide the fax and copier services. This should only be a temporary solution. You will waste too much time, otherwise. A two line phone switch can be installed to answer the home and business lines from one instrument. A headset is very valuable if much time is spent on the telephone. The basic fax machine should have an automatic paper cutter, 10-page automatic feed, speed dialing, and Group 3 protocol.

The portable or cellular phone adds flexibility and movement to leave the home office without missing calls. Voice mail or an answering machine can assist with the answering of the telephone and taking messages to prevent an important message from being missed.

A minimum configuration desktop PC should consist of a 2mb main memory, 386SX based CPU with 40mb hard disk storage

and 1.44mb floppy disk. VGA monitor prices are reasonable. A laptop PC is a necessity for traveling use. Compatibility with the desktop PC is a requirement. For any remote access, a 2400 baud modem on either PC is a minimum. The price on 9600 baud v.32 modems is falling rapidly.

A minicomputer is very affordable if it is needed. A small HP3000/37 can be purchased for about the price of a PC. If you do your own support, it is a very inexpensive system.

The printer requirements will determine the type and number of printers. They range from inexpensive portable inkjet to dot matrix to laser page printers. A laser printer is the preferred choice for quality, speed, and low noise level.

Furniture can be obtained from family sources, used office furniture stores, computer stores, and large furniture outlets. The basic office should contain a desk with a good back supporting chair. The standard desk may be too high for comfortable access to a PC keyboard. A computer table should be used instead. A thorough analysis of work flow will determine the placement of equipment. If other individuals visit the office, sufficient seating should be available for their use.

A sufficient supply of office materials can be supplied from the supply room of the company. You will need pens, pencils, paper, disks, and any other items used in the office.

A complete set of manuals are needed for any hardware and software system used. Some software packages offer the option of printing manuals from the programs themselves.

Flexible storage must be abundant and easily available. Shelves can be wall mounted or located in closets. A four drawer file cabinet is needed, also.

SOFTWARE: Basic software considerations are operating systems, word processing, spreadsheet, graphics, data base, communications, language compilers, and electronic mail. Ensure that any software purchased is widely used and supported by all equipment you have.

The usual operating system for the PC is MSDOS but UNIX is gaining in popularity. If you purchase an HP3000, current FOS and subsystems tapes should be provided. Windows for the PC provides a graphic interface to isolate the user from the operating system. A mouse should be used with Windows to derive the most benefit from the interface.

Word processing software needs are driven by the business focus. The needs will vary from simple letters to programming language text to document processing. The popular products are WordPerfect, MS Word, Wordstar, and Page

Maker.

Many people use the Lotus 123 spreadsheet product. There are several compatible products available. Compatibility with other software and hardware is important.

Graphics capability are included in some of the spreadsheet and word processing software. Harvard Graphics provided a very simple way to make the slides for this presentation. It will also make a slide show with your slides for a moving presentation if you use an overhead projector display or a PC projector. High quality graphics are essential to a professional presentation.

There are many data base products available. The best known is DBase IV. Most are relational with menu-driven interfaces. If a fourth generation language is used with the HP 3000, such as Speedware/4GL or Powerhouse, there is the capability of using them with a PC as well as the HP 3000.

Communications products must be chosen based on the systems to be accessed. If access to an HP3000 is required, then Reflection and Business Session are the most popular.

Requirements for programming will determine the language compilers needed. Microsoft has Cobol, Basic, C. Micro Focus has a main frame Cobol. Borland has Pascal and C.

Electronic mail systems vary from simple bulletin boards to extensive products like HP DESK/HP MAIL.

OFFICE ENVIRONMENT: The room used as the office is very important to the telecommuter. Location, power, air conditioning, lighting, and communications are major parts of the successful installation.

A room shared with other home activities will function temporarily but should not be considered permanent. The ideal is a spare room, such as a bedroom or basement is preferred. To limit noise and other outside interference, a room with doors to shut out the rest of the house activity should be used. A room could be added on the house if the funds are available and the existing structure will support it.

Power for all system devices should be wired directly to dedicated outlets with 20 amp breakers. Air conditioning, if required, should be provided to maintain a proper temperature level year round. Lighting should be sufficient to ease eye strain. Communications lines should be available for voice and data communications. Voice/fax/modem switches are available to share lines to cut cost.

WORK ATTITUDE: A successful telecommuter must act as if he is at a formal office. Mood is very important. You need to set work hours that are realistic. You must be a self starter because no one is there to manage your time and work

habits. Also, you must resist the temptation to be a workaholic.

You should dress based on the interaction with others and at least have some basic formality, not a bathrobe. You will tend to wear out tennis shoes and blue jeans instead of suits and dress shirts.

You will find a need for social interaction. You don't have the office water fountain to congregate around. This could cause large phone bills to satisfy the urge to socialize.

The support that is given by the family is most important to your success. They have to agree to the fact that you are at the office for specific times during the day even though you have some flexibility. Stay involved with your local users group.

EMPLOYER POLICIES: The employer can save costs by having you telecommute. You need to have a written agreement for expense reimbursements, attendance at meetings, time and progress reporting, capital equipment purchases, and insurance. This agreement provides the formal ties to the employer. It also delineates responsibilities. This is important to the corporate employee who works part-time or full-time. A corporate employee should have a business plan for their telecommuting operation. This helps set priorities and provides economic justification, if needed.

HOW TO TELECOMMUTE AND RETAIN YOUR SANITY

5135-7

OUTLINE

- DEFINITION AND JUSTIFICATION
- EQUIPMENT
- SOFTWARE
- OFFICE ENVIRONMENT
- WORK ATTITUDES
- EMPLOYER POLICIES
- SUMMARY

5135-8

DEFINITION AND JUSTIFICATION

I

- DEFINITION OF TELECOMMUTER
- TELE – Far off, distant, remote/at, over, from, or to a distance.
- COMMUTER – A person who travels regularly between two points at some distance.

5135-9

DEFINITION AND JUSTIFICATION

II

- CORPORATE EMPLOYEE
  FULL TIME
  PART TIME
- JUSTIFICATION
  AVAILABILITY OF TALENT
  PERSONAL SITUATION
  TRANSPORTATION RESTRAINTS

5135-10

EQUIPMENT

OFFICE MACHINES

- TELEPHONE
- FAX
- COPIER
- CELLULAR PHONE
- VOICE MAIL

5135-11

EQUIPMENT

COMPUTER SYSTEMS

- DESKTOP PC
- LAPTOP PC
- MODEM
- MINI COMPUTER
- PRINTER

5135-12

EQUIPMENT

MISCELLANEOUS

- FURNITURE
- SUPPLIES
- MANUALS
- STORAGE

5135-13

SOFTWARE

I

- OPERATING SYSTEM
- WORD PROCESSING
- SPREADSHEET
- GRAPHICS

5135-14

SOFTWARE

II

- DATA BASE / 4GL
- COMMUNICATIONS
- LANGUAGE COMPILERS
- ELECTRONIC MAIL

5135-15

OFFICE ENVIRONMENT

- ROOM
  SHARED, SPARE, ADD-ON
- POWER
- AIR CONDITIONING
- LIGHTING
- COMMUNICATIONS LINES

5135-16

WORK ATTITUDE

- ACT LIKE GOING TO OFFICE
- SELF STARTER
- SET NORMAL HOURS
- DRESS CODE
- SOCIAL INTERACTION
- FAMILY SUPPORT

5135-17

EMPLOYER POLICIES

- EXPENSE REIMBURSEMENT
- MEETINGS
- TIME/PROGRESS REPORTING
- EQUIPMENT
- INSURANCE

SUMMARY

- DEFINE BUSINESS
- PROPER ENVIRONMENT
- PROFESSIONAL ATTITUDE
- PROPER PLANNING

5135-19

The Truth About Purchasing HP Computer Equipment

By: Charles H. Finley, Jr.
ConAm Corporation
546 N. Oak Street
Inglewood, CA 90302
(213) 419-2200

What you are about to read may scare, enlighten or possibly help you. One thing is clear, however, it will give you a better understanding of what's really happening in the Hewlett-Packard (HP) equipment market. There is so much competition, price cutting and plug compatible equipment that today even the most knowledgeable person is confused. We hope that this article not only stops the confusion but provides you with the basic knowledge to properly evaluate the best equipment source for your requirements. Our first step is to explain the most desirable features one should seek in a computer equipment vendor.

What Are You Really Buying?

If you're like most people you're not just interested in buying a piece of hardware, you're interested in purchasing the **use** of the hardware. Also, presumably, after you've determined the need for the equipment and gotten approval to purchase it, you need the use of the equipment as soon as you can get it. In other words, you want the use of it as soon as possible. To an HP user, use of the equipment also means a few other things. These include eligibility of HP maintenance, reliable operation, good appearance and good service in the event that there are problems before the equipment is covered by a maintenance contract.

Costs or Savings

Most of us hate to overpay. Some companies enjoy a volume discount arrangement with HP due to their size and/or the volume of business they do with either HP or an authorized value added reseller. This discount varies as a function of product type and volume of purchases. The maximum end-user discount is 28% for most larger items and 38% for smaller items, such as PC's. Another potential source of savings is used equipment. Used equipment is available from a number of sources, including HP and VAR's. Savings in this arena are available to any company regardless of their purchase volume.

Quick Delivery

If you need the use of the equipment, obviously, the sooner you get it the sooner you can put it to use. HP manufactures most equipment to order. Therefore, quick delivery in most cases means purchasing the equipment from a source other than HP.

Guarantees and Warranties

Most HP users arrange for a hardware support service agreement either with HP or with a third party maintenance vendor. Not all equipment is eligible to be placed under these contracts. In order to qualify for these agreements, the equipment must not only work reliably, but it usually must have been at an acceptable maintenance revision level. If the equipment is not at a sufficiently current engineering level, someone will have to pay to have it upgraded.

Clear Title

Most computer buyers believe that if they purchase equipment from someone and pay them for it, they own it. Many user's have been very disappointed to find out that a bank in Topeka really owns it. Things get really interesting when the bank wants it back and the broker it was purchased from has already spent the money. It usually doesn't pay to sue someone with a net worth of less than the amount of the damages.

Purchasing Flexibility

In many cases, customers have equipment that they no longer need. The most obvious case is when either a system, component or sub-system is being replaced by a newer model of the same type. This is called an upgrade. HP, unlike most manufacturers, has a very generous trade-in program. Therefore, if a user wants to get more disc space by trading in their older model disc drive for a newer model, HP usually has a program in place to accommodate this need. However, sometimes the trade-in equipment is worth more to another user than it is to HP. In this case, the owner is better off selling the unwanted equipment to a third party rather than trading it in. Also, if the user has an unwanted printer, and they need more ports for their computer, HP cannot offer a trade-in. This is another situation where the third-party market can accommodate the user.

Customer Satisfaction

The first point we established is that most people are interested in purchasing the use of hardware. Having equipment that doesn't work delivered to you quickly is almost as bad as not having it at all. Even new equipment will sometimes fail. However, it is extremely important that failed equipment either be replaced or repaired in a timely manner. This usually requires a financially sound source who is in a position either to pay for repairs or to replace the equipment quickly. The number of failures in the field can be greatly reduced if the vendor tests all

equipment before shipping. There ain't no free lunch, however. Testing and guaranteeing cost more than not testing and guaranteeing.

The Costs of Testing and Refurbishing

New equipment is tested as part of the manufacturing process. HP "remanufactured" equipment is brought to a central site and repaired and tested. However, very few sources for third-party equipment have the resources or the knowledge to test or refurbish equipment. The resources needed are substantial. In addition to the trained technicians, one needs manuals, test equipment, support personnel and facilities. All equipment must be brought to the testing facility. Naturally, this implies that there are freight costs. Moreover, there is the carrying cost of having the item in inventory until it is sold.

The Total Cost of Purchasing

Acquiring the use of computer equipment is not as simple as it seems. The most obvious component is the actual cost of the item. In the case of a new item or an HP remanufactured item, that cost, plus any time lost in productivity if the equipment fails, represent the entire cost. All costs associated with repair or replacement of the faulty equipment are borne by HP. Equipment failure of a new disc drive might require a reload, for example. This requires not only MIS personnel involvement, but also denies the use of the computer system to the enterprise, thereby causing a loss in productivity. Equipment purchased from third parties can sometimes have the same cost characteristics as equipment purchased from HP, though this is rarely the case. Usually, when equipment is purchased from third parties, there are additional expenses, such as freight and installation. (HP charges a minimum of $560 for installation of equipment purchased from third parties.) In many cases warranties from third parties either are non-existent or are worthless. Therefore, if the equipment breaks, the buyer ends up picking up the tab. This, as some users have found, can be very expensive. So expensive, that after all repair charges, the equipment costs exceed the price of new equipment.

The Certificate of Maintainability Myth

"When you believe in things you don't understand, then you suffer, superstition ain't the way . . . " Thank you Stevie Wonder. There is a document that is available from HP called a "Certificate of Maintainability." Superstition has it that it will protect the buyer of used equipment from all evil. According to modern mythology, if an item covered by an HP Certificate is found to be defective within 90 days of when the certificate is issued, HP will repair it at no charge. However, there are so many exceptions to this that it is fairly simple for HP to find that the item in question is not covered. If HP is paid to supervise the packing and loading, the protection is a little better. However, it's still not complete. The only complete protection available is from the company the

equipment is purchased from. In many cases the Certificate is something a disreputable broker hides behind while the previous owner, the new owner and HP resolve the issue. The most likely company to pay for the repair is the new owner, not HP. A responsible third party will protect the buyer in the event the Certificate is not honored by HP.

How Prices Vary

If you purchase the equipment you need from HP or a VAB at new prices, this will usually be the most expensive way to purchase the equipment. There is, however, another way to make your purchase expensive, that is, to purchase it from the wrong third party. Even if prices appear to be lower, look at what is provided with the sale. Sometimes, particularly in the case of entire systems, freight and/or installation are included by one vendor and not by others. In other cases, particularly with very old items, a spares kit is sent along in case there are problems at the time of installation; however, most vendors do not do this. If a vendor does provide spares, it is a valuable extra that is worth both time savings and money to the buyer. Sometimes, particularly, if the third party is not very familiar with the equipment, the wrong cable or part is all that is available. In many cases the device may not be in good repair (i.e. worn heads on a tape drive). Purchasing equipment from a third party is not inherently risky. However, if you try you can make it so. If you're already saving a tremendous amount by purchasing from a reliable third party, why do something risky by selecting a marginal source in order to "save" $200 or a few percentage points.

How Much To Pay

Now that you understand the different considerations in buying computer equipment, choose the type of vendor that best fits your requirements. Price should be the last consideration. It's silly to make a risky purchase for the sake of saving a few dollars. You should choose the best vendor who will most likely fulfill your business requirement at a reasonable price. The reasonable price may not be the cheapest price.

Where and How To Buy

There are at least eleven ways to buy. The first is to purchase the latest models new directly from HP, the manufacturer. Here you have the advantage of always getting the latest models and support directly from Hewlett-Packard. According to Datapro, HP is consistently rated the best among large manufacturers in user satisfaction. However, everything has a price, and in this case you will usually have to wait longer and pay more than you would if you chose one of the other options.

HP Remarketing

HP is probably the largest lessor of HP manufactured equipment. All leasing companies have departments and/or divisions that dispose of returned previously-leased assets. At HP that entity is called the Finance and Remarketing Division or FRD. FRD does not sell directly to the ultimate customer but works through the HP established sales channels to get the returned items to the customer either as a sale, a rental or a lease. Equipment sold through FRD is refurbished to the extent that they refer to their refurbishment process as remanufacturing. It is extremely comprehensive. They say that they routinely replace about 70 percent of the mechanical parts in computers and peripherals.

All equipment that is processed through FRD, which is supposed to include rental and leasing returns, demonstration machines, trade-ins and asset sales (internally used computers) is remanufactured before it is sold. The latest engineering changes are applied and the equipment is thoroughly tested. It should, therefore, come as no surprise that HP remanufactured equipment is the most expensive way to purchase used equipment, because remanufacturing costs a lot and someone needs to pay for it.

FRD believes that it is HP's policy to sell all used equipment through FRD and to remanufacture it. However in the real world we have observed a different set of policies which, of course, allow for different opportunities for the buyer.

HP Demo Equipment Sales

HP does sell used, non-remanufactured equipment. The most common way this is done is through the local sales office. This demonstration equipment is equipment that is used in the local sales office. Demo equipment is seldom mentioned on the first sales call by your HP sales representative. If the salesperson is asked about it, the customer is then made aware that, understandably, only a limited quantity of demo equipment is available, that it is difficult to free it up and wouldn't the customer prefer to have new equipment anyway. Competition has a way of easing the availability of demo equipment. Our firm helped HP find at least three demo Series 950's last year. Countless Series 70's have also been found that way. The full range of HP hardware is available as demo equipment from terminals to Series 950's. "Ask and you shall receive."

HP Fixed Asset Sales

Hewlett-Packard is perhaps the largest user of HP manufactured computer hardware. This should surprise no one. Also, just as other companies must dispose of assets previously used internally which have become surplus, so must HP. As mentioned previously, FRD is usually responsible for disposing of surplus assets. However, on at least one occasion, a San Francisco Bay Area manufacturer saved more than $90,000 from HP's published remarketed prices by purchasing an assemblage of used equipment assets from HP. The equipment came from four different divisions! What is particularly remarkable about this is

that the buyer paid only about one third of HP's published remarketed prices! As far as I know, the equipment was not remanufactured, and it did not come with the usual HP warranties. I don't even know whether it was sold in a manner consistent with HP policies.

The VAB

Value Added Businesses (VAB's) are authorized resellers of HP equipment who have arranged to purchase equipment from HP at a discount in exchange for stimulating the sale of HP equipment to end users along with some added value, usually software. VAB's usually sell the same items that HP sales reps sell, and that is a combination of new and remarketed (remanufactured) equipment. HP frowns on VAB's who sell used equipment not purchased directly from HP itself. Most VAB's seem to provide small discounts to customers on hardware. However, because their normal source of equipment is HP's established distribution channel, lead times are usually a problem.

The Dealer

A used equipment dealership is a company that buys and sells computer equipment, actually takes possession of the equipment, places it into inventory, performs at least some cleaning, testing and refurbishment; and, when the equipment is sold, warranties it and provides clear title to the buyer.

If the buyer's primary interest is in getting the use of the equipment quickly and at a reasonable cost, the dealer has a lot to offer. Dealers stock equipment, therefore on-time delivery is not as much of a problem as it is for brokers. By taking possession of the equipment, cleaning, testing and refurbishing it, a dealer increases the likelihood that the equipment will not only work at the time of installation, but will also continue to work reliably. It also allows time to improve the appearance of the equipment, a little paint does wonders. Dealer prices and broker prices are usually very close. In fact many times a broker will purchase the item you want from a dealer and mark it up to the end user.

Hewlett-Packard makes unusually reliable hardware, with few exceptions. But even HP delivers some defective units to its customers. Also, even if equipment has been maintained by HP, is packed and transported properly, there is still the possibility that it will not work when it arrives. This is the nature of equipment in general. The best we can hope for is, that if there are problems, they are handled promptly and professionally. This is something a dealer should be able to do. The combination of staff engineers and customer service personnel, and replacement inventory make it possible to cure problems quickly. In many cases, faster than HP can do it! For example, many times a defective terminal, teleprinter, or memory board, etc., can be replaced by a dealer overnight. This is an advantage that having inventory provides.

Trade-in's are an important advantage a dealer offers. Because they are interested in acquiring inventory for resale, they are most likely to allow you to use it as partial payment. Where HP or a VAB have severely restricted trade-in rules, this is not so with a dealer. For example, the only trade-in's allowed by HP are a disk drive for a disc drive and an SPU for an SPU. In other words, one is only allowed to trade-in "like kinds" of equipment. A dealer will take a surplus Series 70 in exchange for Spectrum memory or a printer in exchange for a terminal, just to name a few examples.
Clear title is something few of us worry about. We assume that, if we pay for something, no one can take it away from us. This is totally false. If you purchase equipment for which the seller does not have clear title (i.e. a bank or leasing company has a lien on it), the legal owner can come in and remove the item almost immediately, even if it has crossed state lines! Further, the only legal remedy available to the buyer is to attempt to get reimbursed from the selling party. If the selling party is a broker with a net worth of $5,000 and the computer was worth $150,000, guess what. Your employer will probably explain the problem to you as they watch you clean out your desk. On the other hand, a company with a net worth of one or two million dollars will be so interested in your getting a good title that the problem probably won't arise in the first place. A real dealer must have a substantial net worth. It is required to run that type of business.

As a footnote, remember that even new equipment purchased from a VAB does not automatically come with a clear title. HP will not transfer title until the VAB pays HP. Should the VAB go out of business between the time you pay them and before they pay HP, HP is justified in seeking payment from your firm. If you are not prepared to pay, they can legally seize the equipment.

The Broker/Liquidator

Brokers are different from Dealers. Generally, a broker will only buy equipment that a specific customer has requested. If they do actually purchase equipment for inventory it is usually sold as a liquidator would sell it, that is, all purchased equipment is sold in a very short period of time (i.e. 30 to 60 days).

There are companies that specialize in brokering and liquidating many different brands of computer equipment. These are Multi-vendor Broker/Liquidators. In order to put this in perspective, imagine what would happen if your HP programming and systems staff, the exact same number of people you have now, had responsibility for several other systems that operate on other brand computers. Those same programmers, operators and analysts would have to deal with Wang, Prime, Microdata, Honeywell and Data General operating systems and hardware all at the same time. Could they do as good a job as they do now? It is doubtful that they could. Also imagine if your HP Customer Engineer had the responsibility to learn as many different types of hardware. They, too, would probably not be able to do it.

Jacks-of-all-trades have the same problem in any field. It requires a great deal of resources to master them all.

The smaller broker/liquidator is usually a one to four person company that operates with very little overhead. They rarely own much more than phones, desks, a copier and a facsimile machine. Personnel is usually restricted to the salespeople and a receptionist/secretary. The equipment they sell usually comes directly from its former owner, sight unseen by the broker. Occasionally, these small brokers will liquidate a system in the same fashion as their larger counterpart. However, most of the time, the liquidation is to another broker or to a dealer. Sometimes it is possible to save money by purchasing through a broker, but usually the same equipment is available at almost the same price through a dealer.

Brokers, particularly smaller ones, usually require substantial deposits, regardless of the credit worthiness of the buyer. They use your money as a down payment on the equipment they are buying to satisfy your requirement. What would happen if the seller of the equipment (end user) goes into bankruptcy before the broker gets the equipment out? How do you get your deposit back?

Unless you can save a lot of money, there is no advantage in buying from a broker, but there are several disadvantages. The equipment is almost always sold sight unseen, therefore, it might not even be the correct item. Even if the equipment is brought to the broker/liquidator's site, they seldom have the proper staff to test and refurbish the equipment. It will probably not have been cleaned or tested. Moreover, if there is a problem at the time of installation or during the "warranty" period, how long will it be before the item is repaired or replaced. Finally, if there is some sort of legal problem, such as a title problem, what recourse do you have if damages exceed the broker's net worth?

Auctions

This is the most risky way to purchase equipment. Auction sales are as-is sales. If the equipment turns out to be junk or costs a lot to repair, you're still stuck with it, you have to pay for it in advance. They occur infrequently. However, savings can be substantial.

End User to End User

This can be one of the riskiest ways to purchase HP equipment, and at the same time, the most economical. What makes it risky is the lack of guarantees of the equipment working, the lack of a good warranty, and the title problems. If these purchases are arranged properly, however, they can be a good deal for both buyer and seller. The question becomes, is the amount of savings worth all of the trouble. That's a question only the buyer and seller can answer.

The Listing Service

Listing services are available for selling used equipment. A fee is usually charged to the buyer if one is found. The fee is normally about ten percent. However, the listing service has no real responsibility in the transaction. Considerations such as freight damage, maintenance eligibility and clear title and payment are the responsibility of the buyer and seller. In essence, this form of buying and selling is not very different from an end user to end user transaction. Except that in this case someone collects a fee for getting the two parties together. Very little equipment is bought and sold through this channel.

Computer Stores

Computer stores were selected by HP to sell PC's and certain usually low-end peripherals. They are rarely any more qualified than a broker to do anything else. All of the things that apply to brokers usually would apply to a computer store if you are looking for equipment that they don't normally sell.

The Best Way To Buy

The HP equipment market is confusing. Aggressive telemarketing by almost anyone who has a mailing list and a telephone, quoting almost any price the listener wants to hear, have added to the confusion. I have tried to clear up many of your questions and help you understand what is really happening in the HP equipment sales arena.

There are other issues one should consider in buying computer equipment. Such issues as conversion costs, training, additional software, relicensing costs are only some of the issues one must consider. Because of those issues and various relationships between people and companies, there is no absolute best way to buy. Even though purchasing the latest model new from HP is usually the most expensive option for the smaller end user, it is in many cases the one that makes the most sense on balance. In fact it could be the least expensive option for a large user with a hefty volume discount. Similarly, if a broker/liquidator calls with a phenomenal deal at a time you need it, that could be the best choice, providing you can reduce any risks to an acceptable level. However, there are a large number of instances where your needs could be met by any one of the eleven possible choices. It is my conviction that in those cases, the best way to buy it is from a dealer.

INFORMATICS IN THE FUTURE

Charles Finley
CONAM CORPORATION
546 N. Oak Street
Inglewood, CA 90302
U.S.A.

INTRODUCTION

It seems that not even a week goes by without a significant announcement that affects users of Hewlett Packard (HP) computers. These announcements both from HP and from other hardware and software vendors promise some new outstanding feature or capability. Some of these announcements provide options for enhancing the computing environment we already have, and some offer to replace what we have with something entirely new. Should you migrate to an HPPA environment and if so then to which model? Should you replace what you have with UNIX or a PC network? Should you switch to a relational database management system? These are questions that an increasing number of us must answer. We must all implement our own strategies to manage this change.

This paper contains my perceptions on where the industry is going, in terms of

- Open Systems

- Corporate Processing - Mainframe/Minis/PC's

- Distributed Processing and Networking

- HP MPE/XL and HPUX what the future holds

- Data bases - future developments

- Computer User Groups

OPEN SYSTEMS

An open system is a computing system whose software architecture adheres to defacto and dejure standards. The operating system is some variation of "standard UNIX" and the data communications and networking protocols are also various flavors of standard. The data base can be accessed using the SQL constructs.

Open systems promise independence. The theory is that, if the user is able to purchase the "same" product from a number of different vendors, the price of computer hardware will be less in the long run. A second promise in the open systems idea is that more software will become available to the user at a lower cost. The third Open Systems promise is standardization for all computers used for all applications (on-line transaction processing, engineering, real time data acquisition,etc.) Lets analyze these goals.

First lets look at the idea of cheaper hardware. When you buy a Hewlett Packard computer what are you really buying? It's not just hardware and systems software. You also want service and support. For proof of this look at the PC marketplace. Low end PC's that require little or no support have become commodities. More and more they are being purchased through discounters for the lowest possible price. It is also possible to purchase PC networks the same way but usually the result is not nearly as good. Networks are more complex than simple standalone PC's for this reason many end users are finding that they need to purchase them from firms able to provide adequate service and support. Most end users lack the sophistication to deal with UNIX based system hardware and software problems, as a result they tend to rely on either in-house or outside computer experts for service and support. The point is, if you're going to effectively use the computer you will need help with hardware and systems software. If you are not paying the manufacturer you're either paying someone else or you're not getting it.

Now lets look at the idea of lower cost software. Again lets look at the PC world. The good news is it is possible to get excellent applications software at very low prices in many cases. In order to see the down side lets examine one of the most important factor that has made the HP 3000 so successful, consistency. Computer professionals moving from one HP 3000 installation to another do not have to choose the spooler they are going to use, it comes with the system. All like systems have the same spooler. This has also been true of the data base management system (Turbo IMAGE), and the screen handler (V/Plus), etc. This "standard" environment has allowed software developers to spend most of their time enhancing their products rather than adapting them for different environments. The UNIX user has a choice of at least 50 spoolers and who knows how many data base management systems. Support in that environment is more difficult as you add applications each implemented in its own unique way and utilizing its own database management system, etc. An increase in difficulty ultimately can lead to an increase in support cost and in the learning curve required for newly hired employees if an in-house support staff is used. On top of the increased support cost, there are other costs associated with the required purchase of additional software necessary to run additional applications. Valid licences will need to be required for each unique data base management system, etc.

Finally, there really is such a thing as a jack-of-all-trades and a master-of-none. As with other things meant to be all things to all people, UNIX is not optimized for the same things that MPE is optimized for. MPE was designed for on-line transaction processing in a commercial data processing environment. It's very good at that. It would be a bad choice for use on a real-time system or an

engineering workstation. On the other hand, UNIX will never be as good at on-line transaction processing as MPE is.

There is little doubt that Open Systems will ultimately replace proprietary environments in the long run, its just a matter of time. Most companies believe that the benefits far outweigh any disadvantages. The responsible Information Systems professional will know where his/her service and support is coming from, however, **before** the system is purchased.

CORPORATE PROCESSING - Mainframe/Minis/PC's

Traditionally, mainframes have offered the highest maximum transaction processing rates and allow the largest data bases. So called super minis can now process at what were mainframe rates only a few short years ago. Also the maximum amount of disk storage available on minis seems to increase every month. Therefore the classes of data processing tasks that only mainframes can handle are getting smaller every day.
Similarly, the distinction between minis and PC's (super micros) is also rapidly decreasing.

The PC used as a workstation instead of a terminal has many advantages. One of the most commonly sought advantage is that by using a PC the user is able to perform both functions best done on a PC (spreadsheets, word processing, etc.) and well as have access to the mainframe. Additionally, the concepts of Client/Server and Cooperative computing have extended this advantage.

The PC is also ideally suited to various dedicated, mostly stand-alone functions. For example alarm systems, auto-cad, building environment monitoring.

Economics are the key to predicting the future uses of these various levels of technology. Usually, the newest level of technology is the least expensive to own and operate. If there is a data processing task that can be solved with use of either level of technology, in the long run companies will choose the newest level of technology. It would therefore seem that as minis become more powerful they will replace mainframes and as PC's be come more powerful they will replace minis. Moreover, as newer ways of using PC's become available such as NEWAVE, the ability to use PC's in ways that are impractical for minis or mainframes will make PC's the only choice in many cases. This becomes most obvious in the Client/Server and Cooperative processing environments.

DISTRIBUTED PROCESSING AND NETWORKING

Departmental computing is a form of distributed processing wherein instead of an organization making use of a centralized mainframe for processing several small to mid-range computers are used. These smaller computers serve the needs of particular departments and are linked to either a mainframe, a large mini and/or to other departmental computers. These networking links allow users to continue

to access corporate-wide databases and applications. It also allows the sharing of expensive peripherals such as high speed laser printers and plotters.

There are several advantages to this approach; it saves money on computer hardware while putting computing resources closer to and more under the control of the user department. As end users become more and more computer literate, they are insisting on more autonomy in their use of computers. The availability of networking products such as HP's NS/LAN combination and industry standard protocols such as TCP/IP have made this approach even more practical than at any prior time.

User demand for applications cause application solutions to be implemented on a variety of stand-alone systems and PC networks. It then becomes a challenge for the Information Systems department to find a way to integrate this "solution" with the existing applications. This trend will no doubt continue to gain momentum.

HP MPE/XL AND UNIX - What the future holds

First a disclaimer. I was not an Assembler language diehard. Even though I was proficient with IBM Assembler, as soon as it became practical, I switched to COBOL. I did not hate my mother, and I am not a Communist. Also, no one can prove I said that thing about Richard Nixon, there are no reliable witnesses.

I believe that the HP 3000 with Turbo IMAGE is a better environment than a UNIX based platform with any database management system. There, I said it, and you can't make me take it back.

Still, in the long run UNIX/SQL may ultimately replace the HP 3000. However, for the next few years, I feel that users who already have the HP 3000 will continue to add applications. I also feel that many companies who do not now have an HP 3000 will choose the HP 3000/Turbo IMAGE solution because it does such a good job. Moreover, with the availability of relational database management systems on the HP 3000 and the POSIX compliant addition as well, I believe that even some companies already in a UNIX frame of mind will continue to develop for the HP 3000 environment for quite some time.

DATA BASES - future developments

In the past most of the debates about data base management systems have centered around the relational/hierarchical access methods and implementation schemes. It would seem that the relational access approach is winning in terms of popularity. I think that it is possible and even desirable to implement a database using Turbo IMAGE, a hierarchical method, and access it both in the traditional IMAGE way as well as with a relational front end. The advantage to this approach is that it allows for the most efficient physical implementation of the database while still allowing an industry standard, relational access.

The most interesting issues in mind for the future of data bases have to do with new ways in which they will be used. The concepts of Object oriented and Distributed data bases are the most interesting new concepts. Also interesting is the idea of networked access to data bases.

What makes Object Oriented data bases so interesting is the new classes of applications they will allow. Users will not only be able to retrieve whole documents in new and useful ways but will also be able to store and retrieve voice information. This opens up a whole new world of applications development possibilities.

Distributed data bases are data bases that have parts in different geographic locations. Many large companies will be able to significantly enhance their competitiveness by use of these.

Network access to databases whereby applications can interoperate over multiple platforms and even operating systems are a reality. Oracle for example works on the HP 3000 and 9000 as well as MS/DOS, OS/2, various flavors of UNIX as well as other operating systems. The popular ideas of Client/Server computing and Cooperative Processing will continue to make this form of database access more and more common.

COMPUTER USER GROUPS

Change can have many side effects. It can be interesting and exciting but it can also be frustrating and scary. Much of how change affects us depends on our response to it. For the individual professional, keeping in step with new and emerging technologies is the first prerequisite for managing change and for that matter career advancement.
Certainly, companies that hope to prosper in the coming years must provide their workforce with the tools needed to adapt to change. Conferences sponsored by computer user groups, such as this one, serve both needs. They offer companies and individuals in the computer user community an extremely affordable opportunity to stay up-to-date technically.

Two factors will make computer user groups more important in the future, they are: the rate of change in the computer industry and the reduction of service and support by computer manufacturers which was discussed earlier. That the rate of change will be a factor has already been demonstrated to all of us. We all see how difficult it is to not only stay up-to-date but to not be fooled by products that seem like a good idea at the time but are obsolete in a short time. UNIX computers are now becoming commodities. There is rampant price discounting. In fact a recent article in the Wall Street Journal pointed out that most major computer manufacturers are loosing money on their UNIX product lines! We have all seen evidence that HP is reducing it's service and support levels in order to stay profitable. As manufacturers provide less service and support networks of professionals will become more important.

In his famous book about change called "Future Shock", Alvin Toffler says that "... information is the key ..." to managing change. I thank you all for the opportunity to share my thoughts on the how the current new technologies will affect our the HP user community. I hope that these thoughts stimulate and supplement your thoughts about these subjects. However, the most important idea I feel I must communicate is that the key to success in the Information Industry is not knowing it all at any particular point in time, so much as its is knowing how to stay informed.

Paper Number 5138
Support Contracts - A User's Perspective

Michael M. Finn

21st Century Systems Group
12745 Woodford Way
St. Louis, MO 63044
(314) 298-9659

INTRODUCTION

This paper discusses the major concerns regarding support contracts. For the sake of brevity, details will be introduced during the session.

There are many types of support contracts. The more common types are those that cover hardware, peripherals, operating systems, utilities and applications. This paper considers support contracts for purchased applications from the client's perspective.

Since most of us are not lawyers, we cannot render an opinion on the legality of a contract. That is best left up to general counsel. That is their job.

In the past decade, there has been a tenfold switch in the percentage of in-house versus purchased applications. With that type of growth and opportunity, software providers have developed a multitude of new applications. In turn the need for support for those applications has increased proportionally.

Support is merely another term for service. As the saying goes "A product is only as good as the service behind it", well it goes something like that. The point is, for a purchased application, we must have enough support to maintain and operate the application efficiently. With good support our job will be noticeably easier. With poor support the opposite is true and our job may be impossible.

WHO SHOULD PROVIDE SUPPORT

Who needs support? The answer is, of course, everyone. The question is really a matter of who should provide the support. The providers available are the software vendor, a third party, and internally.

Consideration must be given to the source of purchased software. Was it from the original developer or a third party reseller? If the software we are using was purchased from the original developer, then our first choice would be that vendor.

If we have a diversified trained staff and source code, there is

the possibility of internal support. However, in today's tight economy and the streamlining of many corporation's staffs, internal support is a luxury. The resources needed for such tasks can be expensive.

Third party support is common with small PC based applications but is rare for others. The computer industry is now coming to the age where third parties are starting to offer limited support on the larger applications. In part, this is due to the high turnover in the support department of software vendors.

We will now consider the areas of concern associated with external support. This type of support would be available from software vendors or third party suppliers.

ADVANTAGES OF SUPPORT

There are several advantages to having a support contract. They include but are not limited to:

System expertise - Since the vendor has the personnel that are experts on their systems this allows us access to this expertise.

Additional resources - With external support we have the ability to obtain additional resources without having to hire and train new staff.

Peace of mind - A by-product of external support is the fact that it will provide peace of mind knowing that system expertise and additional resources are available.

Upgrading - Another advantage is that usually with support, upgrading to the next version of the software is provided at no additional (or a nominal) charge.

DISADVANTAGES OF SUPPORT

Some of the disadvantages are:

Dependance on an outside resource - With external support we are dependent upon the third party to provide us with the assistance necessary to complete our objectives.

Expensive - Support contracts tend to be expensive and are usually based on a percentage of the current list price of the software.

Availability of vendor's personnel - Unfortunately we must depend on the vendors availability to provide us the assistance needed. If the vendor is under staffed, we could be looking at a lengthy lead time.

Inadequate personnel - Some vendors will have so called experts in the various applications that they offer. The support is sometimes better for some products than for others. You also lose the total systems person who could assist you in looking at the whole picture.

EXPECTED SUPPORT

To myself and other users, the support we expect would allow for smooth implementations. The help desk would be available 24 hours a day, 365 days a year. If a problem cannot be corrected with phone consultation or via modem then a member of the vendor's staff would be on-site within the hour. Sounds too good to be true and it is. There are some vendors that provide excellent support and some are less than what we expect.

INADEQUATE SUPPORT

Coping with inadequate support has been a problem since the first time one person purchased software from another. Each side of the transaction has its perception of what is supported and how the support is to be administered. There are precautions to take to decrease the possibility of inadequate support or misinterpretations. If support is inadequate, there are ways to change the support received.

First, contact the manager of support or equivalent and explain your concerns. If that does not achieve the desired results, then try that person's supervisor. If you are still not satisfied, then your next recourse is through the vendor's user group if one exists. Finally if the vendor is listed as HP+ or higher, you may try HP for assistance in correcting the problem. Some users have stopped paying the yearly maintenance fee but that usually causes additional delays in the support they already were receiving.

CONTRACTS AND NEGOTIATIONS

When negotiating the original purchase of the software, you have the best opportunity to negotiate a few items in the support section of the contract. As we discussed earlier, we know what the ideal support contract would contain; it would contain the provisions to allow us to receive our perception of expected support. At a minimum include the following in your contract and negotiations:

Cost - The cost should be clearly specified. Is it based on CPU size, number of users, number of transactions, etc.? Be specific here, many users have been startled when they discover the cost to upgrade to a larger machine. The cost of on-site service if you need assistance, does it only include travel and hourly rates? What about off hours assistance, is it at the same rates if available at all?

Timing - The time required to correct errors should be explicitly defined. "A reasonable amount of time" leaves you in an unwanted position.

Expenses - Be alert to any possible hidden charges. These usually are in the areas of time and materials or other out of pocket expenses.

Consultants - Include in your contract a section that states you have the approval of which consultants a vendor may assign you. This allows you to pick and choose among the available people, thus allowing you to have the most knowledgeable assistance.

Outside Consultants - Be sure that the contract allows you to have third party consultants. They should have complete access to vendor documentation as well as be able to attend all training courses. This allows you the opportunity to obtain the best additional assistance.

Finally, have your legal department review the document. They may be able to point out ambiguities or terms that the vendor can use to its advantage. Be careful out there.

SOME READING MATERIALS

Data Processing Contracts, 2nd ed.
Dick H. Brandon & Sidney Segelstein, VanNorstrand Reinhold,
1984

Computer Contract Negotiations
Joseph Auer & Charles Edison Harris, VanNorstrand Reinhold,
1884

A Software Law Primer
Frederick W. Neitzke, VanNorstrand Reinhold,
1984

Legal Care For Your Software
Daniel Remer, Nolo Press,
1982

Law & The Computer
Michael Gemignant, CBI Publishing Co.,
1981

Software Protection
G. Gervaise Davis III, VanNorstrand Reinhold,
1985

PAPER NO. 5139
SAVING TRAINING DOLLARS
BY
Jeff Franz
21st Century Systems Group
12745 Woodford Way
St. Louis MO 63044
(314) 298-9659

The need for training on computerized applications is obvious, develop the knowledge base of the users to make maximum use of the system. This statement is true regardless of whether the application is purchased or developed in house.

What may not be as obvious is the cost of this training. Costs such as course fees, travel and materials are somewhat easy to identify. Other implicit costs may not. These costs include the trainee's time, use of the facilities and other on site costs. Do not forget to include any post training study by the attenders.

Like all expenditures, the cost of training should be minimized while still achieving the maximum effect. The best way to make the maximum use of these training dollars is to determine your needs and receive the proper training at the proper time. Another way to say this is PLAN.

Planning helps reduce costs in two important ways. First of all it allows you to avoid unnecessary training. Unneeded training is a total waste of time and resources. Secondly, planning allows the users to receive the training they need WHEN they need it. This aspect is often overlooked. The timing is important since skills erode over time if not used. As Aristotle said, "that which is used develops, that which is not used wastes away." If the skills gained in training are wasted, so are your training dollars.

There are four areas to consider when determining the level and source of training. These areas are:

1. The goals of or reasons for the training.
2. The audience (who will be trained).
3. The trainers.
4. The complexity (and cost) of the applications on which training will be performed.

Each of these areas will be discussed in the following paragraphs.

Goals or Reasons

Each person who will receive training is going to have a different need that the training must fill. Although individuals are unique, it is often helpful to group them together based upon their system outlook. The two most common groups in this area are the users and the technical people. There is a further breakdown based upon whether the application is purchased from an outside supplier or developed

in house.

This breakdown provides the axis for a matrix. This matrix consists of four boxes. They are users of purchased applications, users of in house applications, technical for purchased applications and technical for in house applications.

Examination of the reasons for training as well as needs as defined within the matrix will provide a guide toward reducing costs. Like the entire program, it must be used in conjunction with the other parts (audience, providers and complexity) to create a holistic approach to this problem.

New Applications

In this case, the users of purchased applications and the technical for purchased applications are in the same boat. Both are starting from scratch. It is important to get a good start on the applications. The timing of the training will be important since everything is new. A strong grasp of the basics is also essential. Choose the providers carefully since there will probably not be an in house expert or "guru" and you cannot be assured of developing one. Outside providers may be a way of life.

If the application is developed in house, things tend to change. The best providers for the training may be, and probably are, in house (although they may not be the developers of the application). The timing of training, and its pace, may not be as important since the experts are generally available. Remember; however, the implicit cost of the trainers time. It may be advisable to spend some time developing basic training guides to minimize the drain on the experts time.

Decrease Errors and Gain Efficiency

In this area, the users and technical people have different interests. The users will want to know all the "tricks of the trade" for the applications themselves. Users of in house applications may be looking for program changes that will assist them in reducing errors and make the applications easier to use. For this type of training, the providers need to be well versed in operations as well as have some knowledge of the application. The timing for this type of training will be well after initial implementation.

Technical people should be trained for this goal as well as the users. Their interests will lie in providing settings, error codes, etc. to minimize operator or user error. They may also be able to streamline the manual intervention on the operator side. Providers for this type of training may be somewhat generalized in nature and at times almost

theoretical.

Enhance Image of the Department

While on its surface, there appears to be little dollar value in enhancing the image of the department, there can be sound financial reasons for it. Knowledgeable users can have immediate results from in house data processing staffs for developed applications. This can save a lot of time since there would not be as much arguing and retesting of problems found by the users.

Those who license purchased applications can have clout with the vendor since they may be candidates for references or site visits, smart software companies know where their bread is buttered. Further, better informed software clients will have more power with the user groups. This can help further their own organization's interests with regard to enhancements, fixes, etc.

This type of training can have a wide variety of providers since some of this image building can be gained from generalized sources. The attenders will probably be those in supervisory or management positions.

Growth of the Department, Organization or Individual

The growth of the organization depends upon the growth of the individuals within it. This type of training allows the individuals to further their own knowledge base as well as provides a source of outside ideas to the organization. It also can provide a morale boost to the attenders as well as the people underneath.

The providers for this type of training will depend upon the training desired. It can range from lunch meetings to day long seminars even to university courses. The attenders may be clerical (for short seminars) up to senior management.

Audience

Another major consideration in a training plan is the audience. This factor cannot be over emphasized. If the training is provided to the wrong level, it could be lost or even cause resentment by the attenders.

Also the audience will have a lot of say regarding the length of classes and the level of detail. As a rule of thumb, the higher the level, the less the detail. Therefore, a class that might last three to four days for clerical staff, may be condensed to one day for management level personnel.

It is valuable to review the various levels in an organization and determine the type of training that would return the most

value for the money. The levels to be reviewed are; temporaries, clerical staff, supervisory staff, management and executives.

Temporaries - They will need to know how to perform the tasks at hand. There is little need, if any, for background information on the organization or any other matter.

Clerical staff - Like temporaries, they need to focus on the tasks at hand. It may be helpful for them to be somewhat cross trained so they can see the effect their actions have on other areas.

Supervisory staff - They will need to have some detail of the tasks to be performed but not too much. Review and analysis features of applications or generalized control issues will be of interest to this group.

Management - This group will have little need for details of given tasks but will want to take a cursory look at them. Their main emphasis will be on "the big picture". As such, their training should concentrate on data retrieval, summarization capabilities and analysis work.

Executives - These high ranking officials are only interested in the big picture. Their training (or education) should be at the highest level possible. Anything lower may bore or anger them.

The audience plays an important part in saving your training dollars. Careful consideration of the audience when scheduling attenders will assure that the right people receive the right training. Failure to recognize this factor will result in spending more for explicit costs (e.g. course fees) and implicit costs (e.g. management time).

Providers

This is another "who" in the training equation. It is essential that the individual or organization providing the training be the best available. Therefore, every attempt should be made to obtain resumes or references on the trainer. Even though the organization may have a good reputation, it is the trainer that is the direct source. Other items such as qualifying the course for continuing education credits (CPE's) can be real pluses since it may save money in organizations that require accreditation.

There are several sources for training. Each source must be evaluated based upon the particular need of the audience at that time. Some of the more common providers follow:

Outside seminars/Consultants - These provide a good forum

for information exchange and some generalized information may be gained. They also may have a cost advantage since they are done for large groups. There also may be no travel cost associated with them.

Accredited schools - These will help the individuals to a broader knowledge base. The value for the organization may be mainly technical expertise. The main disadvantage is that the course(s) may not come close to the organization's training goals. The other is that course content and quality can vary greatly by instructor.

Software vendor - They are generally a good source of training for their own product. The standard courses may have good materials. The training may be offered on site. This would allow you to send more participants without the added travel cost. A disadvantage to this source is that the instructor quality can vary. It is not uncommon for vendors to use newer employees to perform basic classes. Further, the trainer may be specialized on that particular application. While this is not necessarily a bad thing, sometimes more generalized knowledge is needed in an integrated systems environment. It may pay to check various client references to see if there are independent providers of training on the vendor's package.

Internal sources - The main advantage is probably cost. As will be discussed in a later section, "training the trainer" can be a cost effective way of training large groups of people. For internally developed applications, this may be the only source of training. The main disadvantage to internally taught courses is that the materials may not be as good as those developed elsewhere. Other possible problems include finding individuals to teach the course(s) and coordinating the timing.

Determining the provider of the training is an integral part of saving money. Failure to adequately qualify trainers can result in lost course fees, since no one will receive benefit from it. Even if the course fees are refunded, the implicit costs remain. There is no such thing as free training.

Complexity of the Application

The complexity of the application can enter into the cost/benefit ratio for training. Generally, more complex applications will have different expertise levels. This will lead to greater planning challenges since you must walk before you can run. The question will be, when can you run? Additionally, the goal of the training will be to get more out of the application.

As a side light, complex applications generally cost more. The more the application costs (investment) the more that training is worth (maintenance). Also, the investment in a complex application could be driven by an organization's need. This may mean that a greater level of expertise is required to meet that need.

The complexity of an application brings several more planning challenges. First, what additional training will be needed to make the most of the investment? What is the best time for this training? Where can qualified instructors be found? How can these qualifications be verified? What other organizations have this application and are using it in a similar (if not like) fashion? These questions must be answered before putting together any training plan.

Development of the Plan

The development of a training plan requires a holistic approach. This includes all of the factors listed above. Failure to take each factor into consideration when deciding on training could result in, at a minimum, wasted time and effort.

It is essential that a decision be made, who will learn what, when. This includes clerks all the way to executives. Yes, executives may need to learn about some specific applications. It is important not to over train individuals. It does not take Monet-like ability to paint a room.

The primary focus so far has been on the who and the what of the training equation. It is just as important to consider the when.

It is often said, "Timing is everything." While this may be true in some areas, it is not everything to training. This does not mean that the timing of training can be ignored. It is necessary to consider the four essential elements in determining the timing of training.

The first part is purely monetary. It may be that there is a contractual limit to receive "free" training. The decision here would be whether to take the training or forego it since it may do no good and therefore waste the time of the participants as well as any travel money.

Another component deals with reducing outside training costs by performing the training using in house personnel. When the original training is performed from the outside, this approach is called "Train the Trainer".

This approach can be extremely cost-effective since outside training is often more expensive on a per diem basis. It may also be possible to use some of the materials provided with

the original course thus negating one of the normal drawbacks of in house training. The main disadvantage of this approach is that it takes some time to effect it since the in house trainers will need to become proficient on the application. Further, since everyone will lose some of the original training, the secondary (in house) training may not be as effective as the original.

Losing training brings up another crucial element in determining the timing of training. When will the individuals be able to apply what has been learned?

It is generally known that normal people lose approximately 60 percent of what they learn almost immediately after learning it. This percentage increases dramatically as time increases (remember the inverse squares law?) Therefore, the training should be timed when the knowledge can be applied.

The final segment in the timing area is something to avoid. This is a complete re-train of existing personnel on the entire system. Whatever can be done to avoid this, do it.

Like the other four components of developing a training plan, timing is one that should be taken into account. It shares equal importance with goals, audience, providers and complexity.

Training can be an important investment in the well being of any organization. Reducing the investment cost is a worthy goal especially when this is not done at the expense of reduced training.

Planning the subject, attenders and timing of any training can reduce these costs substantially. Investing time to investigate the providers and content of courses being considered can eliminate unneeded or wasteful instruction, freeing training dollars for more productive use.

So you want to buy a computer system?

Robert M. Gignac
Thomas Lighting inc.
189 Bullock Drive
Markham, Ontario (Canada)
L3P 1W4
(416) 294-9570

Abstract

This paper will present an insider's view of how Thomas Lighting Inc. acquired their current computer system. It might have been entitled "How not to buy a computer in 10 easy steps". Follow Thomas Lighting through their mutli-vendor, mutli-platform search, and see how they arrived at their decision to run on an IBM AS/400.

By the time this paper is due to be presented, you may also have an update on how the conversion went, as it is scheduled for July 26, 1991.

Introduction

It has often been said some of the best literature begins with the phrase "It was a dark and stormy night...". As this story is non-fiction, and somewhat bizarre, I'm choosing to begin with "Once upon a time...". In addition, I must warn all readers the names of the individuals/companies you'll encounter in this journey have been obscured, to a) Protect the identity of the innocent, b) Somewhat shelter the guilty, and c) Ensure I still have a job when I am finished.

The Distant Past (1978-1988)

Once upon a time there was a small lighting company known as C&M Products (known today as Thomas Lighting), a fully owned subsidiary of Thomas Industries in Louisville, Kentucky. They had grown as small companies are prone to do, and at some point resigned themselves to the fact that an automated (ie: computerized) means must be found to control their data. From information I have been able to gather, the year was 1979 when the big push was made. An automated inventory control, purchasing, accounting - or what today might be called "integrated manufacturing" - package was custom written for Thomas by Simplified Operating Systems (S.O.S. for short - pun intended) in Toronto. Perkin-Elmer was the equipment of choice, and after a rocky start, things were rolling. Eventually, C&M outgrew the little 8K machine and it was replaced by a

Data General Mod IV capable of running the same software. In time C&M outgrew this machine as well, as C&M acquired other divisions due to corporate growth in the U.S. and expansion of their product line. Since S.O.S. had long since departed the scene, the proprietary rights to their software had been picked up by Olivetti Canada and their Network Systems Group. At this point in time the move to Unix was on. Olivetti provided C&M with an entry level machine, an LSX 3020 to support the software which was soon upgraded to an LSX 3030. This is the machine we are currently supporting, as we head toward conversion.

The Recent Past (1988-1990)

Sometime in 1988 it became apparent to those at C&M (we became Thomas Lighting in August 1990) that the software developed for the company 10 years earlier when it was a 4 million dollar operation, was getting stretched to its limits by a 35 million dollar operation. The hardware was adequate (and in my opinion still is...) but the software... this is a story in its own right. It was developed in a an extension of Micos Extensive Basic using sequential files, index pointers, and a 16K memory limitation. These factors yielded cumbersome hard to maintain code. Maintenance was made even more difficult when there were no user manuals for the applications and the programs had few, if any, comments imbedded in it, other than the program name, date of creation and the author. Execution speed was also an issue, especially when our product files exceed 35,000 items and we have over 7,000 bills of material, all of our accounting and purchasing functions, not to mention on-line order entry. New software was definitely in order.

An agreement was signed with a vendor to provide a Unix based system written in Progress 4GL which would meet C&M's needs. This integrated manufacturing software would be a new adventure, and it would also include something completely new for C&M - Material Requirements Planning (M.R.P). This is the point where I entered the picture.

I joined C&M in February 1990 after spending 5 years in an HP shop where I had the chance to dabble in Unix and 4GL's on the Motorola Unix boxes our company produced. Implementation of this new software was to fall into the hands of myself and our Inventory Control manager, who had joined the company only two months before I arrived. Approximately two months into the installation and testing of this software we encountered our first major hurdle - the software did not work. For the sake of saving myself from some needless and costly litigation, let me rephrase that - the software did not work in the fashion that C&M required it to work. It was flashy, had nice windowing features, even pull-down menus (these were some of the features which had attracted the president, plant

manager, and the controller to the software - all of whom had parted company with Thomas by the time I arrived). But some of the major features you would think should be in a finished piece of software - ability to create purchase orders, indented bill of material facilities and the aforementioned M.R.P. module - did not exist beyond a selection option on the menu, which when chosen, issued the message "Option currently under development".

Things at this point started to get interesting. Corporate, which had pretty much left us alone until this point in time, suddenly became very interested when two new hires began claiming the software for which they had laid out a substantial amount of money (they had purchased it outright, almost sight unseen) didn't work, and probably couldn't be made to work in a time frame that fit our needs. Clearly, some alternative solutions had to be investigated/considered. After continual prodding from the Corporate I.S. Department and the 'special projects' coordinator, we continued to make a go of implementing the software (I even spent a week at Progress software in Boston learning about application development), but we quietly began to investigate alternatives. We started by looking for products which would run on our existing Olivetti platform using the Progress software which we had purchased. Did I say purchased? Well, that is what I was lead to believe. Unfortunately at this time, I found out that we did own the 4GL software, but all we owned was the 'runtime' version of Progress. While allowing you to run your applications, it wouldn't let you access the source code (to look at, or possibly change), nor would it let you develop any add-in routines to the code, or access the relational database structure outside of the object code already developed. Unfortunately, Corporate decided that spending an extra $20,000 for the full 4GL package would not be fiscally responsible.

To some degree, this was a blessing in disguise. If Corporate wouldn't approve the funding, then we had an out to stop trying to get the software working in our test system, especially when we couldn't access the source to determine just why the things that didn't work, didn't work. As well, it wasn't like we didn't have anything else to occupy our time, we still had a business to run.

"Give it your best shot..."

Faced with what we saw as a no-win situation, we heeded the above advice from Corporate and tried to make a go of it. We opened up 'hot-line' communications direct with the vendor's office, and we had assurances that our questions/problems would be addressed within 48 hours (not solved - but at least we would know they were aware of them). In

addition, one of their support specialist practically lived on our dial-in modem, working on support problems.

Why did we rate all of the this special attention? First, as we found out much to our surprise/dismay, the vendor needed us up and running as much as we did. Although the vendor would probably disagree with my terminology, we were, for lack of a better word, a 'beta-test' site for this release of the software. We found this out by accident by requesting, over and over and over, a customer list from which we could contact people and commiserate over what did/didn't work, and how to work around what might be limitations in the system. I can see why a company might not divulge a list like this during the pre-sales phase of a project, but we had already paid for the software. We did finally get a list, and when we contacted people, the majority of those we spoke with either hadn't purchased the modules we had, or were light-years away from planned implementation. Others were using the modules in some fashion, but were not as complex a manufacturing process as we were. The second reason for all this special attention was the fact two of our sister companies (located in Montreal, Quebec and Gravenhurst, Ontario) were also planning to go with the same software. Since we were the first, if we failed in the process perhaps the other two sites would not proceed in this direction either.

Try as we might (and we did try!) the project just never came together. Our problems at this point weren't just limited to the software. We became critically short on disc space on the Olivetti machine, and the overhead associated with the 4GL software had a noticeable effect on the response time of our production system. If we were really going to give this a shot we required: a) more terminals; b) more memory; and c) more disc space (I know, the dreams of every I.S. Department any of us have ever worked for...).

Olivetti came to our rescue with an attractive option - we could upgrade to their new LSX 3070 processor at no cost upfront (cost would be incurred after successful implementation). Since we could not add more terminals or disc to our existing system in an economical fashion, this seemed the way to proceed. Well, almost. The Progress software, while ported to the LSX series of computers had not been ported to the 3070 yet, and since it used a redesigned processor architecture, the estimated cost to port Progress to the 3070 was an astounding $75,000. Progress didn't want to pick up the cost, our software vendor didn't want to pick up the cost, and we certainly weren't going to pick up the cost for trying something that in the end still might not work. Stalemate. This actually turned out to be the last straw and at a meeting held two weeks after this event, we officially called a halt to the project. We now had some latitude to search for another solution, and Corporate kept whispering "I.B.M."....

Could we go the "HP Way"? - Part One

When it was apparent we needed another solution to our problem we tend to stick to what we know best. I had just come out of a shop after spending 5 years working with an HP machine and while certainly no expert, I knew my way around the hardware and the operating system. Our materials manager had several years experience using MANMAN software from ASK and had gone through a major conversion with his previous employer. Several of our design engineers had worked with MANMAN as well. It just seemed 'logical' to take a look at this option.

After several productive meetings with the local HP representatives and the people from ASK we already had a rough sketch of a configuration and a good idea of the pricing structure. One major advantage we had by proceeding in this direction was the fact that one of our plants in the U.S. was already running MANMAN on an HP. Gardco Lighting of San Leandro, California is running an HP 955, and they also use that same machine to support our Emco division out of Chicago, Illinois. Multiple site licenses from ASK made the pricing very attractive, and bundled together with a model 925 we felt we had a very viable alternative: both functional and cost effective.

We presented this option to members of the Corporate I.S. Department who were up for an analysis/postmortem on the Unix project. The proposal was not well received. Corporate suffered from that, "I don't see I.B.M. in this proposal..." syndrome. Far be it for us to present an effective solution that beat the pants off any I.B.M alternative we had seen proposed by Corporate. Especially when we threw in the fact that our corporation has only one Class 'A' MRP operation, and it just happens to be Gardco, who runs HP/MANMAN. It was at this point that I heard the funniest line I've heard uttered during this entire project - "The problem with HP is that it's old technology". After I picked myself up off the floor, I proceeded to explain to our Corporate M.I.S. director (who at the time of this writing is our former Corporate M.I.S. director) that RISC based architecture, DAT tape backup, fiber-optic device interfaces, and CD-ROM technology does not fit the category of 'old technology'. For better or worse, I also offered to take him out for dinner the moment I.B.M. catches up to HP in this area. I'm still awaiting the phone call.....

Nobody ever got fired buying I.B.M.

As we had proceeded through our trial and error period, some funny (ie: strange) things were happening within the Corporate infrastructure. It had been determined by the powers-that-be that the hardware of choice for any future development within Thomas would be 'Big Blue', otherwise

known as I.B.M. Now, I'm not opposed to standardization (in fact, I believe it is a good thing at times), as long as it is done with some conscious thought and foresight. To this day (despite numerous requests to Corporate), I have not seen one shred of data that supports why they choose I.B.M. over HP, Dec or any other hardware vendor. One inside source at Corporate (sounds like a spy novel, doesn't it?) told me the only real reason they decided to standardize was the fact the Corporate MIS group got 'tired' of seeing request after request for different hardware/software combinations and having to determine if the equipment being proposed was a) powerful enough; b) flexible/expandable for future growth; and c) had approved software that would allow interfacing to other divisions and back to Corporate where required. Some might ask if that isn't part of an I.S. professionals job description?

Unfortunately, standardization does have its drawbacks. As a mutli-divisional company each of the sites within Thomas had pretty much gone out on their own to acquire computer resources (or already had them upon being acquired). Existing equipment ranged from IBM (System 38, AS/400, and 3090 Mainframe), to HP (955) and Olivetti (LSX3030, and LSX3010). Operating systems ranged from MVS, OS/400, OS/36 (I.B.M.), MPE/XL (HP) and OS/X Unix (Olivetti). How do you begin to standardize a group as diverse as this?

As you may have guessed by now, we were the lucky ones chosen to be first and prove that a standardized hardware and software system could be implemented successfully. Since the hardware had already chosen for us (IBM AS/400), at least we would have a hand in choosing the software we would be running/supporting. Wouldn't we? Shouldn't we? From my own skewed perspective, we finally started to do things right. We assembled a team comprising of I.S. staff, materials managers, and accounting managers from all divisions of Thomas and locked ourselves away for 3 days at I.B.M. in Memphis, Tennessee for demonstrations of prospective software. The vendors included S.S.A., Pansophic and H.F.A. Now, spending 10 hour days analyzing software presentations is not my idea of fun, but we were armed with preset questionnaires created using BuySmart software. Furthermore, we had managed to roll each division's unique requirements into one document that was sent to all the vendors in advance. Our goal was to arrive at a consensus on one package to become the standard within Thomas to incorporate into our manufacturing plants.

After three grueling days of bad coffee, stale donuts and sugar-coated sales pitches, the verdict was in. S.S.A. and their BPCS product was deemed superior to the others and even from my own biased anti-IBM perspective, the package was pretty slick. Pop-up windows, on-line help, integrated graphics and relational database technology combined for a fully featured and user friendly piece of software. I was a little wary of

the RPG/400 language it was developed in, but S.S.A offered an excellent case tool package that would aid in any in-house development or maintenance that would be required. All in all, I think everyone left Memphis feeling pretty good about where we were headed.

Let's Spec out the System

Now that we knew which hardware we were going to run on and the software we were going to run on it, the final hurdle was to determine how much machine we would need. We had several meetings with the Toronto S.S.A representatives and determined an AS/400 model B45 would be the proper machine to support Thomas's needs, both current and for some time to come. What started to concern me at this point however, was the physical size of the machine. While not overly large, we were looking at a single rack machine approximately 31" X 42" X 66" and we were faced with putting it in an existing room that only measured 7' X 14' which already contained our Olivetti machine, one line printer, one page-laser printer, a UPS and the desk for the system console. Needless to say, the room was a bit crowded. For some unknown period of time we were going to be faced with running two systems, and I was apprehensive that the existing room would work. Other people had assured me however, that all would be fine. We'll get back to this topic later.

The final configuration was an AS/400 B45 with 24MG of memory, 3.6GB of disc, one 800 lpm printer and 40 IBM 3477 Infowindow display terminals. All equipment to be provided by I.B.M. Or should it be? Much like the HP world, the I.B.M world comes complete with a vast array of third-party supplied devices that often outperform the I.B.M. products at a fraction of the price. By my calculations we could have saved ourselves in the vicinity of $40,000 by purchasing just third-party terminals, add-in memory and 2 line printers to replace the one specified by I.B.M. To a company in an industry already suffering the effects of a recession, this seemed pretty prudent. So we did it, right? Wrong. As you might expect, I.B.M. was not too impressed with my enthusiasm to attach other people's equipment to their machine. Thinly veiled threats of possible service problems, tales of woe about other customers who had done what we considered and lived to regret it, and discussions of losing our discount structure, was I.B.M's reply. In the end, we decided to be a 'true-blue' I.B.M. shop, and I remained the sole dissenter to the decision. We may pay for it in the long run however, as we ended up only purchasing forty terminals and only one system printer, knowing full well that we have forty-five terminals and three printers on our existing system. We also have requests for at least a dozen more from people who don't have one now but who will require one when we get the system up and running. Why?

Because we (like most firms...) had budgetary problems to contend with. We were given a dollar figure from Corporate designating how much we could spend, and to stay below that level required some compromises. It remains to be seen how the terminal situation will be resolved.

Could we go the "HP way"? - Part Two

As we started to tally up the costs for going I.B.M. one thing was clear, it was going to cost a bundle. In addition to the hardware/software costs, we had retraining for the I.S. staff as none of us had any I.B.M. experience, recabling costs as our existing 4 wire/unshielded cable wouldn't support I.B.M.'s terminals, and a UPS to support the higher power requirements of the machine.

The final total was staggering to a firm totally unprepared for this type of endeavor, and even Corporate was having trouble giving us approval for the project funding. We felt if we could find a cheaper solution, perhaps we could end up with what we had wanted in the first place, since money seemed to the the roadblock at this time. We approached our Gardco division and enquired about their ability to support us in much the same manner as they support our Chicago division. A week of phone calls and faxes determined that all we would really need was more disc space, memory and a terminal controller for Gardco, and we would need terminals, printers, some form of communications equipment, and a leased phone line across the country. While we tallied up the costs for all of the hardware involved, things looked promising. The leased line turned out to the major problem. Not in acquiring one, but in paying for it. Canada, unlike the U.S., doesn't exactly have a free market system in the communications area. The monthly cost to get a factional T1 link to San Leandro, California to Markham, Ontario was $7,500. That may not sound too bad, but break it down, and it would have only cost $1,475 from San Leandro to Detriot, Michigan, then $6,025 to go the last 250 miles from Detroit to Markham. Facing an expense cost of $90,000 a year on an ongoing basis (as opposed to the ability to capitalize hardware) brought an end to this option

About this same time, we received word from Corporate that not only were we in the final stages of approval for the I.B.M. project, but we would have the chance to be the first plant within Thomas to put the 'standardized' solution into place.

A slight change of plans...

What we didn't know at the time was that our new hardware/software solution was about to be altered just prior to our signing the contracts. We were all set to sign with I.B.M. and S.S.A. until we got a phonecall from

Corporate informing us that Mapics/DB from I.B.M. would now be the software of choice. Needless to say, this took us quite by surprise (as it did our other divisions). Our immediate concern was that we had never seen the software run, and it wasn't even on the short list of vendors we looked at in Memphis. Why? Because at that point in time there were not enough installed sites of Mapics/DB for it to make the short list. Where have we seen this scenario before?

My concern then shifted to our hardware specifications which had been based on an entirely different piece of software. We met for two days with the software group at I.B.M. to get a feel for the package, and while different than BPCS, we felt that it would do the job for us (remembering all the while that just about anything would be better than what we were using). Our machine as configured would be adequate due to better machine utilization by the Mapics/DB software. On the plus side, Mapics/DB was cheaper than BPCS which brought the cost of the project down. We were apprehensive about where we were headed, but at least we were moving again.

Merry Christmas Thomas...

The deal was finally signed late one Thursday evening in October, with the machine to arrive 7 days prior to Christmas. Two days before the signing, IBM announced a new promotion on their 9332-600 disc drives (they just happened to be the ones we were ordering) and we were able to get seven for the price of six. More disc for no more money is always a good deal. Just one minor problem; the seventh drive caused us to exceed the capacity of our single rack configuration. Now we required two racks, which in effect doubled the physical size of the system. As you'll remember from the earlier discussion, we were hard pressed to support a single rack in the existing computer room, a double rack machine was out of the question.

No more room at the Inn...

What had troubled me a little with the single rack in the old room was really disturbing now that we were getting a two rack machine. After all, nobody had ever allocated any funding for building a new room, as they were convinced the old was was going to work just fine. The search for a new location was on. After three days of furious measuring, planning and assessment we located a spot for the new machine, and two weeks later (once all the departments that had to be relocated had calmed down...) construction was underway. As seemed to be the norm for this project, it was one problem after another. Move the existing sprinkler system out,

move the people out, then our fire-rated windows didn't arrive, then our doors were late, then the lights (our own product no less...) were late. A project that should have taken 5-7 days at the most (after all the room was only 16' x 14' with two walls already in place) took 14 days and was to be ready 5 days after the machine was due to arrive. During the creation of this room the issue of fire protection was raised.

We were/are using Halon in our existing room, and we felt that it would be prudent to run a second tank off the existing panel for the time being, eventually moving the control panel into the new room once we were completely cut over to the new machine. IBM felt this was a good idea. Everyone was happy. Well, almost.....as we were to find out 3 months to the day that we made the decision, our Corporate insurance company informed us that Halon by itself would not be acceptable fire protection for our new machine. We must install a secondary sprinkler system in the room for protection as well (I then checked the Corporate M.I.S. guidelines manual for further direction under the topic of 'system protection', but as you may have guessed, I found no such information in print). As I am writing this (April 7, 1991), we are currently getting quotations to remove our ceiling and reinstall the piping for the sprinkler system that we had painstakingly removed three months earlier. This should be a lot of fun as the machine is already in place and we will have to disable the Halon system to work in the ceiling.

The delay in readying the room didn't turn out to be as significant a problem as it first appeared. Due to some fourth quarter delivery problems from the I.B.M. Rochester plant, our machine didn't arrive in time for Christmas unpacking. Neither did our UPS power supply for the machine. Due to the ongoing conflict in the middle-east at the time, Canada Customs inspectors were going through every piece of electronic equipment with a magnifying glass. Equipment that would normally clear customs in 2-4 days took 11-14, putting a crimp into our plans. Both the UPS and the AS/400 were ready for installation the same day - January 4, 1991.

Let's plug this thing in.....

The early part of the day saw a flurry of activity in the new computer room. One UPS technician, an electrician, 2 IBM C.E.'s (Customer Engineers), a Bell telephone representative (to install our outside phone lines), and myself trying to stay out of everyone's way, and still keep track of everything that was happening. By 2:00pm we were ready, time to plug everything in and throw the switch. Would it work? I asked the question aloud by accident and the I.B.M rep overheard me. It turned out to be a blessing. "We can check the UPS power output if you'd like to be

doubly sure...", so I agreed. As nervous as I was about turning on the machine and seeing nothing happen, it seemed like a good idea. The AS/400 runs on 208 volt single-phase power, with one plug for each system rack. All the equipment in the racks are powered off it. As the voltage meter lit up I smiled, we had power, single phase power at that. That was until, "I think we have a problem here...." shattered the moment. We had 222 volts out of one plug, 231 from the other. The AS/400 is a very picky machine, susceptible to power problems (that's why we needed the UPS in the first place, no internal battery such as HP provides) and I.B.M wouldn't turn on the machine with this power input. Forty-five minutes later I'm refereeing a discussion between the UPS technician and the I.B.M. reps over how the machine will/won't run on that power and how the UPS is/isn't working properly, and how the I.B.M. reps have someplace else to be, so can we please get on with solving the problem. I convince the UPS technician to fix the problem (which I shouldn't have had to, it's their job...), now we are finally ready to throw the switch.

We plugged the racks into the UPS and pushed the red IPL (initial program load) button and we were off and running. I must have driven the I.B.M. guys crazy by asking every 2 minutes, "Does it always take this long to IPL?". It actually only took about 20 minutes (longer today now that we have data in the file systems...), it just seemed longer at the time. Once the IPL was complete I had an active system console, but the rest of the system was unusable because there was no software loaded on it, except for a skeleton operating system. The software would be coming later....

The system is now operable

The software load was completed three days later by our Mapics support group. I found out later that I could have done some of it myself if only someone had written the instructions in English. At least an English that a non-I.B.M.'er would understand. I.B.M., for those not familiar with their terminology, is very acronym driven. That is why I didn't realize that the tapes which I had in my possession (and that had been arriving at the rate of 3-5 a week since we signed the deal) included the operating system, our application software, and all of the system utility software we had purchased. But it isn't as easy a loading a tape and restoring the files. Once loaded, you then need to check for outstanding PTF's (program temporary fixes), then acquire the correct PTF's to bring you up to the current release level, or you have the choice of applying only those PTF's that you feel apply to your site (otherwise known as 'taking life into your own hands'). Our system became fully operational on January 11th, and the education process began in earnest.

Studying manuals, as people in the computer field are aware, is a time consuming task and is probably second to none in terms of inducing sleep. It is a fine balancing act to try to learn about a new operating system, new application software, and trying to attach all of your peripherals to a new machine using connectivity devices that, until 3 months earlier, you never knew existed. Balance all of this off against keeping your existing software running, your existing users happy, and dealing with the backlog of requests that can't be solved with the line "Just wait until the new software arrives, ok?". The education and conversion processes are being tackled in earnest as I am writing this and they are not what this story is about. For that you will have to wait until next year. What I'd like to do at this point is my own personal postmortem on the process we went through that got us to this point.

Things 'right', 'wrong' and otherwise

As with all things we undertake, reflecting upon them once completed will yield some interesting results. There will be moments of brilliance, moments of confusion, and moments where you'll wonder how you ever made that decision. This section will attempt to examine the major areas relating to the project and determine how we arrived at those decisions, and include my own opinion on the correctness of the decision.

Software

It would be easy enough to say a mistake was made when the Progress based Unix solution was chosen, but I wasn't around at that point in time and I don't desire to implicate others. In the time honored tradition of many firms, we didn't necessarily want the 'correct' solution, we wanted a solution we could afford, which would salvage some of our existing investment, and which would make life better, if not great. This is how I believe we ended up with a 'runtime' version of Progress rather than the full 4GL package, a move which severely hampered us when the project started to falter. The initial concern was to save the investment in the Olivetti hardware, and find an affordable solution. The interesting thing was that even after the package had been chosen, many of the people in key areas in our company (purchasing, accounting, order entry) had never seen the software. Some never ever did see it before we removed it from the system, once the decision to stop the project had been made.

This same mistake was not repeated in the search for I.B.M.-based software, as key people from all areas of the company, across all divisions were included in the selection process, and everyone had a say about their likes/dislikes in what they had seen. This process cost the

company some significant dollars to perform, and they are to be commended for it. The curious issue is that once the decision was made (ie: BPCS), it was overridden at Corporate. One might have been tempted to ask, "If you didn't want to hear my opinion, why ask for it?"

On the plus side though, once we were moving towards Mapics/DB, we made sure all of the various departments here at Thomas had input into critiquing the package. This highlighted both some software limitations and some procedural inefficiencies; issues which are being addressed as we proceed through the testing/conversion phase.

Hardware

I'll dispense with the period prior to the Olivetti, as I wasn't involved, and it has no bearing on what we eventually did. I can understand why we tried to save the investment in the hardware when the project started, but in reality the hardware wasn't that expensive, and support for it was less than acceptable from my viewpoint. Even by going to other hardware, we could have recouped the loss by transferring the machine to one of our sister companies (an option that is still open) as both have decided to proceed with the software that we felt was unacceptable. We could have also used it as a base for our eventual CAD system, something that was always talked about during this project.

When new hardware became inevitable, the surprising issue is the edict that came from Corporate. As I said earlier, if you do the groundwork and prove that one was superior to another, then you have something to base your decision on and defend it. I don't believe this happened in this case. True, the majority of our U.S. plants run on some form of I.B.M. equipment, however, only one had AS/400. Two had HP and the three Canadian sites all had Olivetti. The one criteria I kept hearing over and over was the connectivity issue. I still refuse to believe that we would have any less trouble getting an HP or DEC to talk to I.B.M. than we will getting our AS/400 talking to our plants in the U.S. My personal belief is that in this day and age hardware is seldom the issue, software is the key. We could have found the world's best manufacturing package that met everyone's needs 100%, and if it didn't run on I.B.M. then we weren't going to use it, and I believe that was the wrong attitude to take.

Don't let me give you the wrong impression here. I know for a fact that our implementation of AS/400 and Mapics/DB will yield wonders for Thomas Lighting because it is light-years ahead of where we started from in terms of software, and it will allow us to manipulate our information hundreds of times better than we can with our current system. It is the process in which we got here that is being criticized.

Other Stuff

As is usually the case when a company considers a major computer acquisition, some things got overlooked, other things were missed because we didn't know any better. One issue here is the computer room. We had never planned for it in the first place, so it was a major surprise the day it was announced that we had to find a new place to put the machine. We could have hired a company to design/build one for us, but we choose to piece it together based on ideas/materials we had available to us. We now have a room that is indeed functional, but we made some mistakes, the most notable being the Halon fire system. On the other hand, if Corporate didn't want Halon installed anywhere due to insurance regulations, this should have been communicated via the Corporate I.S. guidelines.

We were totally unprepared for the amount of decisions that we would have to make with regards to the hardware installation. Selection of a wiring system (twisted pair, twinax, or I.B.M. Type One), selection of a UPS (on-line, in-line, or ferro-resonnant), and the decisions regarding third party equipment (all of which I lost). We started out this project as a two person I.S. department fairly confident that we knew how to run our operation, and ended up as a two person department (with one new member...) who at times doubted our sanity about making the transition. If I was going to do it again (and who knows, maybe we will...) I imagine that I would push for getting a consultant in to aid with the process. Even though it would cost some significant dollars, there is no substitute for experience when it comes to a project like this. The other thing we were totally unprepared for was the amount of time to cut the system over. At least the rest of the company seemed unprepared. I'm not sure if there is anything worse than setting an inflexible target date for cutover without ever consulting with the I.S. department about the lead times to complete particular tasks. This is of course, exactly what happened. By the time this paper is due to be presented I'll be able to update you about: a) whether or not we made the target date; and b) how we managed it, if we do.

The other major thorn in the side about this whole project was the relationship between ourselves and Corporate. I don't want to get into a discussion of U.S. and Canadian bilateral trade agreements, but suffice it to say that a 'branch-plant' mentality block existed between how we view what we do here, and how Corporate views what we do here. We had no input into the standardization discussions, as did some of the U.S. I.S. departments, nor were we initially invited to the Memphis software evaluation, we found out by accident about it, and basically forced them to invite us. Working in this type of environment can be difficult, especially when people start to make decisions for you without your consent. Imagine

my surprise when I received a phone call from I.B.M. in Toronto about our initial system configuration which had been sent up from Corporate without any consultation about our processes, volumes, or number of peripherals we would require. Over the timeframe of this project our relationship with Corporate has gotten better, only because we refused to play by their rules on a few occasions, and they started to take seriously the fact that we do at times know what we are doing up here.

Summary

This paper, while in many ways a rambling story, represents a reality that I'm sure many firms face when they attempt to implement a major computer project. Be prepared for the fact that you will never have enough time, money, resources or background information on which to base your decisions. Call as many people as you can for help, and use whatever support networks you have available fill in the gaps. And try to remember that when once the task is completed, it was the excitement about getting the system implemented and running that attracted you to this profession in the first place.

TITLE: Motivation: When "KITA" Won't Work

AUTHOR: Robert M. Gignac
Systems Analyst
54 Crawford Rose Drive
Aurora, Ontario L4G 4RY
CANADA Phone (416) 841-5632

FINAL PAPER NOT AVAILABLE, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO. 5141

5142

Managing the Non-Networked PC

David W. Haberman

Innovative Information Systems, Inc.
63 Nahatan Street
Norwood, MA 02062

(617) 769-7511

In 1977, when I bought my first PC, PCs were virtually nonexistent in the workplace. Over the past 14 years the PC has changed the way we do business. In the early '80s very few organizations had a plan for implementing PCs. PCs were an oddity that were provided to some accounting people in the back room. There was no plan for integrating PCs into the workplace.

By the late '80s PCs were a major fixture in the work environment. Most of these PCs were purchased one at a time by individuals within the organization. These individuals usually learned to use the PC on their own.

The boom in PCs within the workplace created many problems. Since systems were purchased on an "as needed" basis there was very little coordination of hardware/software platforms. There was almost no training available, and support was not available within the organization. MIS managers found themselves getting support calls on 50 different PC programs. All purchased without MIS involvement, never mind approval.

We have now reached a point where managing PCs within the organization has become a major management issue. This paper will discuss the issues involved. These issues include:

- Who is responsible for PCs within the organization
- Coordination of software purchases
- Coordination of hardware purchases
- Managing repairs
- Managing software upgrades

- Hardware support

- Software support

- Training

I joined my organization in 1987. At that time we had 3 PCs, an HP150, a Compaq portable II, and a Panasonic. Like most organizations we had a mixture of hardware and software. We also had no plan to purchase PCs. We basically bought the least expensive we could find. One year, six months, and twenty PCs later we realized we had a problem. We were not taking advantage of economies of scale. We were not making bulk purchases. Supporting 3 different word processing packages was impossible. Coordinating repairs and warranty work between vendors was a nightmare.

As resident "expert" I was coordinating these tasks on an as needed basis. At one point managing PCs and providing support was taking almost 50% of my time. At this point we put together a PC strategy and long range plan. Since then I have discussed the issues involved with many MIS shops and learned I was not alone.

Who is responsible

Who in your organization is responsible for PCs. In asking this question I have received many different responses. They include:

Individual department heads - PCs are generally purchased and approved at the department level. Therefore all coordination of PCs is done by the individual department heads

Purchasing - Purchasing is responsible for all repairs and purchases.

Office administration - The PCs are used for word processing and administrative type tasks.

MIS - PCs are computers

Some guy in accounting - This guy is the resident PC expert who everybody comes to for advice.

Today more and more organizations are centralizing PC responsibility on a PC coordinator. This persons job description usually includes:

Developing hardware and software standards

Coordinating training

Providing software support

Coordinating hardware/software purchases and repairs

At what point do you need a PC coordinator? In tracking my hours spent supporting PCs, I found I spent between 30 to 40 minutes a week for each PC in PC related tasks. At somewhere between 40 - 60 PCs this becomes a full time position. In speaking with PC coordinators I have found that one person can provide support for approximately 80 - 100 PCs at that point the level of support begins to drop drastically.

Coordinating Software Purchases

In looking through many organizations you can find at least four different word processors, 2 different spreadsheets, and all sorts of database and utilities being used. It is important for an organization to develop software standards. An organization with a standard set of software tools can reduce the purchase costs of their software. Most software companies offer site licenses or bulk discounts.

Standardizing on software allows the organization to maximize its training and support resources. Training classes can be developed and tailored to the organization. Support staff can be trained in a minimum of applications. This will reduce the training expense and allow support people the opportunity to gain greater expertise in the chosen platforms.

By standardizing on software you are better prepared to meet future technologies. Imagine a technology that obsoletes all your current software. If everyone is standardized data conversion is easily managed.

In many organizations you ill find many illegal copies of software. This is a dangerous practice. An article in PC Week described a company that paid $70,000 dollars in fines for illegal software. Some disgruntled employees of the company reported the violations to the Software Publishers Association. The SFA came in with federal marshals and audited the companies systems. They found over 200 illegal

versions of software on the systems. The company paid a $70,000 settlement and then had to purchase the software.

You may think 200 versions of illegal software is a lot. If you have 40 PCs and each has 5 illegal programs on it you've got 200 illegal programs.

The PC coordinator must maintain proof of all software purchases and register all software with the vendors. Software companies are constantly upgrading their products. If you register your software you can usually receive the upgraded versions at a fraction of the list price. If you standardize your software and hardware environments you can develop batch programs to walk your users through the software installations and upgrades. This saves the organization time and money.

Coordinating Hardware Purchases

I have heard many times that all PCs are the same. Anybody who owns three different brands of PCs knows this is not true. MS-DOS may be MS-DOS but that does not account for how the PC vendor chose to manage memory. Many vendor build programs into chips. The system set up programs differ from PC to PC. When inserting add on boards many systems have internal switches have to be set.

By standardizing on one hardware platform you can cut down on troubleshooting time. You can also set up memory management and menu batch programs that are common to all systems. This will reduce the time in involved in setting up PCs for end users.

Standardization will also let you define how you will use external devices. Ever try defining a new printer for a program in windows. Now assume that everyone in the organization has a different printer. This can easily involve days of set up time. Simple issues such as parallel or serial interface can amount to a lot of time if you are managing even 20 PCs.

By standardizing your hardware platforms you can also take advantage of bulk discounts. Even if you don't purchase all your systems at once, many vendors will allow you to spread your purchases over a period of time and still take advantage of discount pricing.

Managing Repairs

I am always asked whether or not I recommend extended warranties for PCs. I have not personally found them valuable. Most PCs come with a 90 day warranty. The extended warranty is usually purchased at the same time. For at least 90 days you have duplicate coverage. PCs are all solid state electronics. My experience has been that if there is a problem it will occur within the first few months.

A support contract for a PC is about $200 - $250 a year. A new PC is about $2000 - $2500. If you have 10 PCs and have 1 breakdown each year you can afford to replace the PC.

When you bring a PC in to repair they do not "repair" the part. They replace it. PCs today are so modular in nature that it is possible to cannibalize old PCs for spare parts or buy replacement parts yourself. A well trained PC person can manage this without a great deal of expensive technical training.

Software Support

PC software support is usually a big pain in the neck for MIS. PC users assume that if you program in RPG you must be an expert at Excel. Nothing can be further from the truth. To provide technical support on PC software you have to be trained in each individual package. I am not saying that a full scale vendor training program must be in place. I am saying that the support person must learn each individual package either on their own or through some training course.

Many organizations say they do not need a PC support person. They let the support be performed by department "experts". This is usually a person within the department who is familiar with the package.

What most organizations have no conception of is the amount of time spent on this type of support. This greatly reduces the productivity of the "expert". They can spend a great deal of time trying to solve a problem they are not qualified to solve. This means that the person with the problem and the expert are both unproductive until the problem is solved.

I challenge any organization using this method to ask your resident experts to keep a log on their desk. Ask the expert to record any time spent on PC support. You will be surprised by the results.

Another method I have seen is to let MIS people take turns at

providing PC support. Each programmer takes a turn at the support desk and they rotate on an ongoing basis. The problem here is continuity. If there is a recurring question or problem each individual may solve the problem over and over again. A single support person can recognize patterns and develop global solutions, rather than dealing with the same problem 100 times.

I believe that by evaluating the amount of time spent om purchasing, supporting, and managing PCs within the organization that PC management should be coordinated at a central point throughout the organization.

Paper #5143

Why Computer Professionals Aren't Extinct....Yet

by
Suzanne M. Harmon
Information Systems Consultant
950 Graham Hill Road
Santa Cruz, Ca 95060
(408) 459-0802

The president of the company is on a business trip and buys a computer magazine at the airport newstand. He spends the flight reading articles which may or may not be accurate or realistic. He returns enthused about sharing in all of the data processing management decisions, such as whether to go relational, and if so, which product.

A bright clerk in one of the user departments teaches themself LOTUS and proceeds to develop their department's application (which is in planning for the future). Their success and satisfaction is such that numerous other departments follow suit. Now the same data is being entered numerous times into different LOTUS systems, and you're being called to solve their problems when disks fill up and the like.

Can computer literate users replace the data processing professional with years of experience under their belt? Will users gradually take total control of all their information processing needs, with the only data processing support required being that of a technician who installs and configures equipment and trouble shoots hardware problems, a sort of new breed of operations manager? Should we all be going back to school to study things like water management in drought environment? Or is this simply a continuing trend in the evolution of job roles and changes in the balance of power?

For years, data processing professionals, in the guise of the MIS department, or the DP department, or whatever it was called, had total control over the automation destiny of the rest of the company or organization. The users got their needs met based on priorities, timeframes, budgets, and application designs under the control of data processing professionals. Their only real choice if they didn't like it was to buck the corporate organization and go to outside, or contract, data processing professionals who might be better or might be worse, but still had all the control over the resulting systems. The users didn't even understand their language enough to know whether they were being fed a line or being dealt with honestly. They had no idea what was possible and what was impossible, what things cost, how long things took, what was reasonable. It was sort of like being a tourist on their own in a country where they didn't speak the language or understand the customs or culture. Of course, very few people like that feeling

of powerlessness, of lack of control, of total helplessness. On the data processing professional's side, it was like any other similar situation, some people abused that power and some people didn't.

The first real exposure non-data processing professionals had to "hands-on" computer use was the advent of entering data into interactive systems. I remember the terror most users felt the first time they had to touch the keyboard. Their fear of doing something wrong, of making a mistake that would "break" the system rendered many of them frozen with anxiety. This first step, though seemingly minor, was the beginning of a transfer of power from data processing professionals to the users. I remember being puzzled when I would go into an environment which was very word processing intensive and being amazed at the fact that though other applications were being used interactively, bringing up word processing had been put on the back burner. In retrospect, I wonder if that's because word processing was an area where the users really knew more about it than the data processing people did, and they were all afraid of that change in the balance of power.

Then PCs hit the marketplace. What that meant was that a user could now have a computer without ever needing a data processing professional's involvement. Interesting, isn't it, that a large number of users had PCs at home before they were ever allowed to get them for office use. They could overcome their fears and develop a sense of confidence within the confines and privacy of their own home. Armed with their newfound knowledge and confidence, they could acquire and use PCs in their work environment without ever even having to notify the data processing staff, if there even was a data processing staff. Trying out various extremely low cost yet user friendly applications gave them insight into what was possible with computer systems. Tools like Lotus and dBase even gave them the capability of developing their own applications. They now had the power; the power to get their own computer, the power to purchase or develop their own software, the power of knowing what was possible, what it cost, how long it took.

Now the group that felt helpless and out of control was the MIS group. Suddenly there were dozens of little systems everywhere. No one had a handle on who had what hardware, let alone what software. It was a mystery if sales figures were being entered into the computer once or a dozen times. Users complained about the accuracy of their reports, but half the time the report they were complaining about MIS had never seen before, because it had been produced by someone from their own Lotus system using data they had entered, and/or were maintaining themselves, which might or might not have any relationship to the data in the central system. The balance of power had truly shifted. The number of requests for application system development or enhancement were substantially reduced and were replaced by requests for support of

PC to printer connectivity, network installation and the like. While MIS had been focusing on such standards as program structure and documentation, the standards of what kind of PCs, printers, and word processing software to purchase were out of control.

The entrepreneurs and the hardware and software vendors were quick to pick up on the fact that the products of tomorrow would be products usable by people with little or no data processing experience. It seems like every new product announcement I see is describing a product whose ultimate goal is to put programmers out of business. It's enough to make a person nervous and insecure. But I think the real message is that the role of the data processing professional is evolving, just as it has for the last thirty years.

In the early days there were programmers and systems analysts, operators, data entry persons, and systems programmers. Programmers didn't design systems, they just did what systems analysts told them to, while the systems analysts designed the systems. This evolved into the role of programmer analysts, people who could both design and program. With the advent of fourth generation languages and prototypes evolved system developers, people who would partner with the users to develop and shape a system. Now I think we are evolving into information managers, people who will basically co-ordinate the use of automated information within organizations. These information managers will be a breed of data base administrator, but not necessarily in the technical sense of that word; people who keep a complete overview of the automated information available within an organization and the relationships between the various groups of information. Their function will be the "normalization" of information at the input or capture level, at the storage level, and at inquiry or reporting levels. They will be advisors to users and prospective users of information, helping them develop their "view" of their information requirements and how to best support them. Some people feel that the evolution of Case technology is specifically headed towards support of this role.

Similarly, operators used to mount tapes, change discs, mount the correct paper on the printer and so on. This job evolved to that of the system manager, a person whose hands-on use of the computer was usually doing backups and resolving hung ports, but who was primarily more of an administrator or liaison between vendors, users, programmers and the like. In the future, this role may evolve into more of a network manager, where network means not only PC Lans, but the networking of all kinds of platforms and technology together in the ultimate world of open architecture.

The data entry person of yesterday is now a user, who enters and maintains ther own information. As technology evolves, this will be the peron who feeds pages into imaging scanners, makes intelligent decisions about what to keep and what to index, and wands in bar-coded information.

The systems programmer of yesteryear is the tool developer of today. The person who develops object oriented data bases, imaging technology, tools to obsolete application programming entirely.

The data processing professional is not going to disappear. Doctors, librarians, teachers, administrators, salespeople and the rest of the user community are too busy keeping up with the technological changes in their own fields to have to keep abreast of the changes in information automation technology as well. There still need to be people within any organization who are experts in hardware, software, information storage and retrieval techniques which are currently available and those which will be available tomorrow. But like most other professionals today, the data processing professional must be flexible, eager to learn and change, eager to embrace the technology which is available and which will make the most long-term sense for their organization. Many data processing people are afraid that they will not be able to be successful in these new roles. Their response is to oppose change, to insist on keeping with the roles and the technology with which they know they have been successful. Like dinosaurs, these people will force their own extinction.

Data processing professionals will also have to adjust to users who insist on having control over their own destiny. Users will continue to evolve in their knowledge, their imagination of what is possible, their sophistication in using information, and their demands to have their needs met. Tools will continue to evolve to support this new breed of user, so that they can design and implement the input, storage and retrieval of their own information requirements in an increasingly sophistcated way. Data processing people will no longer be able to surround themselves in a mystique of mumbo jumbo. They will need to develop sharing and respectful relationships with the user community and with the management structure whose budgets will support them. They will not be able to jam their data processing preferences down the throats of the rest of the organization. They will have to have reasonable and believable arguments to support their recommendations based on a thorough understanding of both what is available technologically, and what is needed within their own organization. This will frequently be uncomfortable because it will require an increasing burden of change and learning on the data processing professional. The data processing people who understand that their real merit is in the foundation of knowledge that years of experience and variety of experience has given them and not their specific knowledge of products and platforms will be the ones to fourish in this environment. Those who are afraid of failure when confronted with learning new hardware, software or techniques will have a harder time. Now is the time for all of us to prepare ourselves for the challenges and opportunities ahead by embracing an attitude of openness to creativity, to change, to letting go of our old perceptions of ourselves and our value and taking a new inventory of our assets for the future.

TITLE: A New Generation of 4GL's

AUTHOR: Karen Heater

Infocentre Corporation

7420 Airport Road
Suite 201

Mississauga, Ontario L4T 4E5

CANADA Phone (416) 678-1841

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 5144

Managing Data Processing Costs

by

Michael C. Hornsby

Beechglen Development Inc.
2026 Beechglen Ct.
Cincinnati, Ohio 45233

(513) 922-0509

The 1980's saw most companies and organizations placing a high priority on implementing new systems and applications, motivated by the potential for cost savings or business improvement.

This resulted in an application "building boom" that has revolutionized the way business is done. However, this free- for-all has led to a piling up of new applications on top of old ones. The result is many redundant obsolete systems, applications and data bases, leaving most companies in a position where data processing operational and application support costs have gone through the roof.

Stated in a tactical fashion, the initial army of analyst and programmers have won battle after battle taking city after city. But after each city is taken, a small well equipt force must be installed in the garrison, and eventually the army runs out of forces and equipment to continue the campaign.

Compounding this problem is the onslaught of personel computer applications that has placed added pressure on the market place for analyst and programmers, and has created an environment where these resources are very expensive to acquire and very difficult to keep.

The purpose of this text is to review the typical data processing budget and provide a process identifying areas and strategies for cost savings.

The first step is to categorize operational activities and costs. In most organizations these costs are decentralized and therefore their true magnitude is substantially diluted. Placing costs in their true categories allows for comparison and identification of possibilities for change.

The second step is to redefine who the customers are and what services are being provided. What usually happens is that after a system is installed, it takes on a life of its own, and the people using it and what they are doing with it changes over time. This is particulary true in situations where networks of personel computers have been installed over existing information systems.

The third step is to develop a strategy for changing the role of the operational support organization to the current and future customer applications and services. This is a crucial step to identification of areas for potential savings and and accessing the requirements for new activities and services.

In many shops the role of computer operations has expanded with very little planning to include support for local area networks, personal computer applications, and electronic mail administration.

The fourth step is to quantify the tasks and objectives identified in steps one through three. Skipping this step is the main reason that those resistant to change win. As stated at the onset, in most companies and organizations, management attention is focused on implementing new systems and applications to achieve some cost savings or efficiencies. A project to gain operational efficiency or cost savings in the data processing department will net a shiny zero. A shiny zero is the label given to someone for taking on an impossible project, saving the company large amounts of money but gaining few points with management.

So, the bottom line is that most data processing budgets are bulging with opportunities for cost savings and efficiencies. But ironically two types of self inflicted ailments insure and perpetuate the continuation this state.

First, the automation of purchasing allowed the growth of continuous purchase orders for services. This means that recurring charges are mindlessly renewed without re-evalution of the need or content of the service.

Secondly, computer hardware and software vendors make a large percentage of their overall revenue from the charges for these services. This explains why they are motivated to make the initial purchase of these support services easy, and changing a purchase order for these services very difficult to impossible.

CATEGORIZING DATA PROCESSING ACTIVITIES AND COSTS

A large portion of a company's information services budget is devoted to the purchase hardware and software and thier associated support services. The purpose of this section is to review the types and kinds of services available and to provide a direct methodology for assesing purchasing procedures and auditing support services.

The cost for hardware support varies among vendors depending on the company's warranty philosophy. Some companies such as DEC and IBM offer extended warranty periods after which their support costs typically run higher. Others, like HP, offer shorter warranties, then have a less expensive ongoing cost. This is all

marketing strategy based on a two or three year cost of ownership analysis.

Software support also varies from vendor to vendor. Some companies sell the software and charge 10% to 20% per year for support. Others lease the software with a monthly payment that includes support.

The bottom line is that when purchasing a system, comparing the cost of ownership over the range of one to five years is important to avoid unwanted surprises when the warranty periods run out.

Why are most companies oversupported? The answer is that contracts are sold with the system or software when it is new and the purchaser is an easy mark for fear, uncertainty, and doubt. Once the contracts are in place it is very difficult to figure out exactly which purchase orders cover what and the vendors make their best effort to keep it that way. To insure the continuation of this situation vendors change the product numbers and pricing structures every 18 to 24 months so that even most of their sales and support representatives can't explain the contractual content the same way twice.

What follows is a structured walk through of the System Support Life Cycle, from the pre-sales process of purchasing hardware and software, the post-sales blues, and to the ongoing support relationship.

Finally we will cover a step-by-step process that will teach you how to perform a support services audit for your data processing environment.

Pre-Sales Support

The process of purchasing a new system is usually a frantic cyclic process, and it is very easy to get confused over who said what would work and who would install and support what. It, therefore, is imperative that you keep very careful records of all hardware and software quotes and configurations.

This is especially true for applications software that will require any amount of customization or integration. The analogy I like to use is: purchasing the application software is like buying the land for a new plant. Someone has to plan and manage a project to build and equip the plant before it can be put into production. Likewise, when a new application is purchased someone has to customize, install, train users, and support the software before it can be put into production.

This documentation will come in handy if anything is missing or doesn't work as advertised. It will also serve to document the progress being made at this frantic system management time.

CASE IN POINT:

Joe, a system manager for an HP3000 series 48, has an account review meeting with his sales and software support representatives, and the topic of improving performance of his FORTRAN/66 applications comes up. HP's recommendation was to upgrade to FORTRAN/77 to improve performance.

Joe decides, based on this recommendation, to purchase the compiler upgrade and sends an accompanying letter containing the minutes of the meeting. After months of effort converting from FORTRAN/66 to FORTRAN/77, Joe finds that performance is no better and even a little worse in some cases.

After a second review meeting with HP Joe discovers what they really meant is that if Joe upgraded to a series 925 and he converted his applications to native mode FORTRAN/77 he would have improved performance.

At this point Joe produces a copy of his letter and purchase order and makes a good case for returning his money on the upgrade because he knows that he can't push an appropriation for a new system through.

The pre-sales period is the time to negotiate for free consulting, performance testing on loan equipment, etc. Once the sale has been made, it is too late to bargain for these cost saving prerequisites. A good example is the migration planning assistance consulting required prior to upgrading to an MPE/XL based HP3000. Many customers that made this a pre-sales condition got this service for free while others paid $6,000 to $8,000 for the same service after the sale.

Pre-sales performance testing is crucial because the number of users a particular system can support is highly dependent on the number and type of applications being executed. Using the number of terminals quoted in the sales literature is definitely a way to get undersold and thus face an expensive and embarrassing CPU and/or memory upgrade.

All of this documentation is necessary to keep everyone honest. If you order incapatible peripherals you will be subject to a heavy restocking charge. But, if you can document that the vendor made the mistake, you'll have a good case for a full refund.

The PC revolution has caused computer vendors to become increasingly competitive across their entire product line. PCs and minicomputers have become commodities that are purchased to solve specific business problems. This trend will be amplified as software application suppliers migrate their applications to UNIX based processors. They are highly motivated to do this as it allows them to market thier product on the widest range of processors and largest number of hardware vendors.

The bottom line for computer vendors is that they must compete on the basis of price, performance, and value added services for market share. In the future some Harvard professor will dub these events as the "COMPUTER WARS OF THE 1990'S".

Most sales organizations have a sales support budget. In HP's case this is an annual budget that applies to sales support consulting , sales expenses and consignment equipment. The annual year starts in December and ends in November. This means that when bargaining, you should be able to get more pre-sales support early in the year and good deals on consignment equipment late in the year.

Purchasing consignment equipment is a good way to get slightly used equipment at a discounted cost. The drawback is that you have to be flexible about when the specific item you want comes up for sale.

When considering the purchase of a new system, it is best to get pricing and support information on a two-dimensional basis. The first dimension to work is the multi-vendor approach. It is a simple fact that those who look at multiple vendors when considering a purchase will be able to bargain for and attain the best overall deal including on loan equipment, programming and consulting. This takes advantage of the sales representative's competitive spirit and leverages the best buy for the dollar spent.

The second dimension to explore is the vendor's OEM channel. Most vendors have extensive OEM or value-added-sellers of their equipment. The arrangement is usually based on a set of credits for systems sold by the OEM. This means that certain OEM's may be in a position to sell you the identical new system for much less than the vendor's list price because they get substantial volume discounts. This is especially true of the large wholesale and retail personal computer chains.

Another important aspect of the pre-sales process to be aware of is the bait and switch or low ball quote technique. This practice is widespread and what happens is that you get hooked on a low price for a low end or underconfigured system and then go through a quote/re-quote cycle several times, ending up with a well padded quote that is much greater than the initial figure.

CASE IN POINT:

Laura, an accounting manager for a medium sized manaufacturer of automobile parts is looking for a system to replace their PC based general ledger and financial applications. With her initial inquiries, she has comparable quotes for an IBM AS400 and an HP3000 in the $35K to $40K range. After selecting a vendor she finds that after each successive meeting the price tag goes up finally reaching above $97K on a formal quote.

This technique can be easily countered by the two-dimensional approach, by requiring formal quotes throughout, and carefully reviewing the differences between the initial quote and each successive quote.

Had this been done in the above case, Laura would have found that the original system had a cartridge tape, dot matrix printer and a single disc drive. And the final quote had a 6250bpi reel-to-reel tape drive, 600lpm printer, and two disc drives.

Consequently, add-on memory, terminals, printers, and personal computers should be evaluated separately. These items can easily be used to pad out a sale, in that it is relatively easy to include more expensive memory and terminals than required, faster printers than required, and certainly overconfigured PCs. All of which can be allotted on the same internal budget but negotiated for and purchased separately when actually needed.

It is amazing how many dollars get tied up in expensive peripherals at the beginning of a project that become obsolete before they come out of the box or any applications actually go into production.

When considering any system, personal computer or peripheral for purchase, it is a wise idea to follow the trade press. This is where you'll find information on soon-to-be released products that may make the items you are considering obsolete, or replace them with identical function at a lower cost.

Post-Sales Support

The most interesting time in any system purchase cycle is the post-sales blues. This twilight zone can range from slight depression to absolute paranoia. The commitment has been made but nothing is happening yet. There are some very important activities that must occur in this period. The siting of the system must be prepared and arranged. The issues of power and air conditioning must be resolved by now or else the reliability of the system after it is installed will be placed in jeopardy.

One of the more interesting problems of purchasing a computer system is receiving it. The system and peripherals will actually arrive over an extended period of time from many locations and in many boxes. It will be difficult to impossible for you to determine if everything you have ordered has arrived. The best procedure is to have a secure place to store all of these pieces and to notify your sales representative that parts of the system have started arriving. Do not open any of the boxes and do not lose any of the shipping papers. It is very helpful if someone takes the time to alert the shipping department that many boxes, some big and some small will start arriving and where to put them. These precautions are necessary because if a part is missing and a trace of the shipping records show that the carrier delivered it, you may end up purchasing the same part twice, or at least wasting a lot of time looking around for it.

The hardware includes installation and a warranty from the date of installation. For HP equipment this warranty period is 90 days. Part of the way into this warranty you will receive a quote for hardware support. It is your responsibility to insure that the equipment is operational after it is installed.

The details regarding software installation are more cloudy. Many software products require a complex configuration effort prior to being operational. These details should be determined in the pre-sales phase so that there are no surprises at installation time, and you don't get stuck paying a conulting fee for an activity you could/should have gotten as a prerequisite to the sale.

The hardware support quoted during the warranty is HP's standard four-hour response from 8AM to 5PM. This does not mean that someone will arrive to fix the problem in four hours, but merely means that HP will respond and acknowledge

your problem within four hours. The actual time it takes for someone to arrive depends on your distance from where your CE is stationed, the availability of spare parts, and the number of other calls placed by other customers at the same time.

The common situation is that the customer will issue a single purchase order for all of the hardware support purchased and thus everything will be on standard four-hour support. In order to mix levels of support, you must issue multiple purchase orders. This is an attempt to make it more complicated and more difficult to mix support coverage.

The other types of hardware coverage available are basic next- day service, time and materials service, and you can also add extra hours of coverage to the standard four hour 8AM to 5PM service.

If you require service during an out-of-coverage period, for example, 2AM, you can pay an uplift charge to get a CE in sooner. As will be discussed in the section on support contracts auditing, having one level of coverage for everything usually means that you are overspending for hardware support.

Whatever level of hardware support you purchase, after the sale your CE will become your primary contact with HP. This is simply because your CE will be spending much more time with you than your sales representative or your SE.

The thing to keep in mind in dealing with your CE is that they aren't clairvoyant and therefore must debug a hardware problem in a very methodical fashion. This procedure at times may be very frustrating because on the surface it is a try this, try that, try this again methodology.

As with any support personnel, you will get much more for your dollar of support if you are fair and keep good lines of communication open. Anyone can make mistakes, but you have to balance out the service you have received over time. And, of course, customers that write letters to document the good and bad points about the service they receive always get the best support.

This is because computer companies tend to be very voice orientated. Just think of how many letters (not surveys or form letters) you have ever received from your computer vendor. So, the net result is that letters from customers carry a great deal of weight, and probably the greatest fear any support person has is a Dear John Young letter.

Unlike hardware support, software carries no warranty period, and if you read the fine print in anyone's software license agreement you'll find an explicit disclaimer of any software warranty. In fact, the main content of software support is updated versions of the software that have enhancements and bugs fixed.

Software support charges range from 10% to 20% of the purchase price per year. HP offers three levels of software support, Basic Line Support (BSL), Response Line Support (RLS) and Team Line Support (TLS).

Assistance is also available on a time and materials basis. The hourly rate is kept at a premium rate to assist in the sale of support contracts. You'll find that local consulting rates range from 50% to 70% of HP's rate depending on your location and the availability of consultants.

Software Support Content Matrix

Content	Team Line Support	Response Line Support	Basic Line Support
Local Support	Included	Time & Materials	Time & Materials
Telephone Support	Included	Included	Time & Materials
Materials Support	Included	Included	Included

BLS provides software and manual updates so that you can keep the MPE operating system up-to-date. It is important to note that vendors usually don't stock manual updates so if you miss or lose a manual update, you'll usually have to repurchase the entire manual set.

RLS includes BLS and adds remote problem-solving via an 800 number. TLS includes RLS and BLS and adds a local SE to perform account planning.

Your software support requirements can be determined by the amount of custom application development that occurs at your shop. If you have purchased a turn key package then all of your support problems and issues can be routed to your software supplier and BLS is all that you require. If you perform your own application development, then RCS may be useful in problem-solving.

The content of TLS support is a number of support management reviews (SMR) and software release planning sessions. In these reviews the SE usually spends the majority of the time selling new hardware and software upgrades and add on's. The number and frequency of these reviews depends on the size of the system under support.

Most sites also have software support contracts for applications. The cost and content of these agreements varies widely. Included in the next section is a methodology for determining the value of these agreements.

Vendor Management

The best method to maximize the value received from both pre-sales support and post-sales contract support agreements is to employ a consistent set of vendor management techniques for all hardware and software vendors.

This process is not as simple as it seems because the priorities change over time as sales and support issues evolve over time. This is because software and hardware vendors usually have different groups of personnel involved in the pre-sales process to get the sale and another group of contract personnel to install and support what was sold.

Thus, we arrive at the first rule of vendor management:

CASE IN POINT:

Art, the best vendor manager I ever knew kept a prioritized list of every problem and issue that his company had with each vendor. At each sales meeting he would get an update on the status of each problem or issue prior to discussing any new products. The result was that the sales force was motivated to get enough management commitment to get the problems resolved in a timely fashion.

The Support Contracts Audit

The support contracts audit is a procedure to determine the contribution that support costs add to your overall data processing budget. The overall goal is to document the software and hardware products on support and assess the benefits of support versus the risks assumed without the support.

All data processing shops fall into one of two catagories, those that kept the appropriate records of their purchases and those that don't have a clue as to what equipment they own, let alone the support products that cover them.

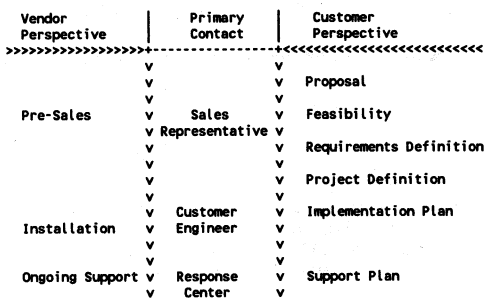
The very first place to start is by building a list of the equipment installed, its location and primary user. This list is important for several reasons, most importantly, disaster and theft insurance. If you think it is a lot of work identifying equipment now, try and picture it after a fire swept through! Or, especially after it grew legs and walked out on its own.

The only way to effectively deal with disappearing equipment is to send out a memo that states what is specifically missing including serial numbers and that if not found within 48 hours a police report will be filed. Perhaps including a reminder notice that theft of company equipment is grounds for immediate termination and that prosecution to the full extent of the law is company policy.

A good way to get most of the equipment inventoried is to initially send out a memo asking each person to list their own equipment.

When completed a sample list could look like this:

Document everything! Make sure to have a written proposal and a written reply from each vendor to the proposal. This is the only way to avoid finger pointing when things don't work later. In most pre-sales environments, no one from the vendor's company will verbally volunteer information that they know will kill the deal. The best way to structure these questions is in the form of a structured proposal.



The second rule of vendor management is: Channel responsibility for all relations with the vendor through one person. The reason for this is that sales representatives will seek to sell at the highest level in your company, especially if they can't meet the requirements of your proposal. The motto is: if you can't beat the proposal then find a way to beat or get arround the author of the proposal.

This becomes increasingly important when moving from the pre-sales to the implementation phase. This is when any problems or variances between what the vendor said they could do and what actually works arise. These must be documented, prioritized and communicated to the vendor.

The third rule of vendor management is: Understand your support arrangements and the corporate structure/culture behind them.

Most large software and hardware vendors have field offices that are organized into separate divisions and organizations with total separate management structures. If you follow the press you'll see that it is very popular for computer companies to reorganize and change these divisions around on what seems like an annual basis.

The net effect is that these different organizations sales, software support, and hardware support have different objectives and motivations. The sales organization obviously has the most clout and the most short-term motivation.

The fourth rule of vendor management is: Communicate with the vendor on a consistent, timely, and appropriate basis via every channel possible.

AMERICAN DEVELOPMENT INC.
COMPUTER EQUIPMENT INVENTORY

Item	Description	Serial Number	Location/User	Ldev #	Cable Id
A1060AH	HP3000 SERIES 935	2839A11217	COMPUTER ROOM		
9133H	20MB DISC HP150	2512A13672	X463		
9133H	20MB DISC HP150	2512A14498	X462		
7980XC	1/2" TAPE DRIVE	2816A11323	COMPUTER ROOM	7	
7937XP	571 MB DISC	2816A20264	COMPUTER ROOM	4	
7937H	571 MB DISC	2709A02861	COMPUTER ROOM	1	
7937H	571 MB DISC	2709A30257	COMPUTER ROOM	2	
7937H	571 MB DISC	2709A30275	COMPUTER ROOM	3	
700/92	TERMINAL	2740A11806	X479	121	B09
700/92	TERMINAL	2740A11850	X480	120	B08
700/92	TERMINAL	2740A11924	X478	122	B10
700/92	TERMINAL	2740A11946	X477	123	B11
700/92	TERMINAL	2740A71945	X476	124	B12
700/92	TERMINAL	2815A11194	X475	125	C01
700/92	TERMINAL	2815A22265	X474	126	C02
700/92	TERMINAL	2815A22276	X473	127	C03
700/92	TERMINAL	2816A23610	X472	128	C04
700/92	TERMINAL	2824A16523	X470	130	C06
700/92	TERMINAL	2824A16584	X469	131	C07
700/92	TERMINAL	2824A26957	COMPUTER ROOM	100	
700/92	TERMINAL	2824A27570	X471	129	C05
700/92	TERMINAL	2838A05503	X468	132	C08
45979D	VECTRA ES	2812V08420	X451	148	D12
45979D	VECTRA ES	2812V19152	X452	147	D11
45979D	VECTRA ES	2812V25182	X450	149	E01
45945D	VECTRA ES/12	2825A25312	X442	157	E09
45945D	VECTRA ES/12	2825A25424	X447	152	E04
45945D	VECTRA ES/12	2825A25427	X446	153	E05
45945D	VECTRA ES/12	2825A25429	X445	154	E06
45945D	VECTRA ES/12	2825A25431	X444	155	E07
45945D	VECTRA ES/12	2825A25434	X443	156	E08
45945D	VECTRA ES/12	2825A25438	X441	158	E10
45945D	VECTRA ES/12	2825A25466	X449	150	E02
45945D	VECTRA ES/12	2825A25474	X440	159	E11
45945D	VECTRA ES/12	2825A25642	X439	160	E12
45945D	VECTRA ES/12	2825A28419	X448	151	E03
45945A	VECTRA MODLE 45	2741A56666	X464	135	C11
45940A	VECTRA MODLE 59	2716A13421	X458	141	D05
45940A	VECTRA MODLE 59	2716A13640	X455	144	D08
45940A	VECTRA MODLE 59	2716A16336	X454	145	D09
45940A	VECTRA MODLE 59	2716A21027	X453	146	D10
45940A	VECTRA MODLE 59	2716A24532	X457	142	D06
45940A	VECTRA MODLE 59	2716A24576	X456	143	D07
45935A	VECTRA MODLE 35	2505A11548	X466	134	C10
45935A	VECTRA MODLE 35	2505A11549	X465		
45935A	VECTRA MODLE 35	2505A11554	X467	133	C09

AMERICAN DEVELOPMENT INC.
COMPUTER EQUIPMENT INVENTORY

Item	Description	Serial Number	Location/User	Ldev #	Cable Id
45849A	HP150 TOUCHSCREEN	2521A17271	X463	136	C12
45849A	HP150 TOUCHSCREEN	2521A17271	X461	138	D02
45849A	HP150 TOUCHSCREEN	2521A17353	X460	139	D03
45849A	HP150 TOUCHSCREEN	2521A17510	X462	137	D01
45849A	HP150 TOUCHSCREEN	2521A19984	X459	140	D04
45816A	20MB DISC VECTRA	2552A33093	X466		
45816A	20MB DISC VECTRA	2552A33094	X465		
45816A	20MB DISC VECTRA	2552A34034	X467		
3630A	COLOR PRO PRINTER	2728A51905	X454		
33440A	SERIES II PRINTER	2652J12674	X467		
33440A	SERIES II PRINTER	2652J29233	X466		
33440A	SERIES II PRINTER	2805A41849	X465		
2686D	LASER JET PRINTER	2604J04215	X455		
2686D	LASER JET PRINTER	2604J04248	X454		
2686D	LASER JET PRINTER	2604J04248	X458		
2686D	LASER JET PRINTER	2604J04289	X456		
2686D	LASER JET PRINTER	2604J04402	X457		
2628A	TERMINAL	2405V13424	X498	102	A02
2628A	TERMINAL	2405V65847	X497	103	A03
2628A	TERMINAL	2426V56842	X496	104	A04
2628A	TERMINAL	2429V56497	X495	105	A05
2622A	TERMINAL	2342A83123	X499	101	A01
2602A	PRINTER	2419A11706	X462		
2601A	PRINTER	2427A14244	X463		
2564B	600 LPM PRINTER	2836A16946	COMPUTER ROOM	5	
2564B	600 LPM PRINTER	2836A26957	COMPUTER ROOM	6	
2563A	300 LPM PRINTER	2428A14077	COMPUTER ROOM		
2392A	TERMINAL	2603A36148	X494	106	A06
2392A	TERMINAL	2603A94018	X488	112	A12
2392A	TERMINAL	2603A94027	X493	107	A07
2392A	TERMINAL	2603A94028	X491	109	A09
2392A	TERMINAL	2603A94029	X490	110	A10
2392A	TERMINAL	2603A94033	X483	117	B05
2392A	TERMINAL	2603A94058	X486	114	B02
2392A	TERMINAL	2603A94067	X484	116	B04
2392A	TERMINAL	2603A94069	X481	119	B07
2392A	TERMINAL	2603A94072	X482	118	B06
2392A	TERMINAL	2603A94893	X492	108	A08
2392A	TERMINAL	2603A94903	X487	113	B01
2392A	TERMINAL	2603A94904	X485	115	B03
2392A	TERMINAL	2603A94999	X489	111	A11
2345A	DIST TERMINAL CTL	2813A12573	COMPUTER ROOM		
2345A	DIST TERMINAL CTL	2813A12574	COMPUTER ROOM		

System Hardware Contracts

The next step is to match your internal equipment list to the various hardware support service contracts you may have. The only way to do this is by the serial numbers from both lists.

The first things that should fall out from this list are any items that are on hardware support but don't appear in your inventory. Usually these are older devices that have been replaced and disposed of, or are in some closet gathering dust. Unfortunately, these older items are also the most expensive to keep under support.

CASE IN POINT:

Melissa, a system manager for large aerospace manufacturer, heads up a project to replace the company's aging 2628 word processing terminals with personal computers. She doesn't find out until twelve months later that the 2628 terminals are still under support, even though they have been sold to a used equipment dealer.

The next level of investigation requires a "replace if broken" decision for each line item on the inventory. As devices age and are replaced by newer-faster-better products, there comes a point where having a service contract is a waste because you would replace the broken device anyway.

Older printers and terminals make good examples of this principle. If you had 2602 daisy wheel printers, or any 264X terminals in use and they broke, you would probably want to replace them with newer technology anyway, so having these types of devices on support is a waste of money. In a lot of cases, you can pay for the new replacement in only one year of support charges for the old device.

A very common situation is that inventories of spare terminals are built up as they are replaced by PC's. Thus, if you have a few spare terminals in inventory, it doesn't make sense to have any of the terminals under support, as you have replacements on hand.

At this point we have decided which items need support and which ones do not. The next decision level is: how much support? After the hardware was purchased during the warranty period, the vendor will make it very easy for you to place the entire system on their standard 8am to 5pm four-hour response. And because the system is new and you don't know any better, you send in a single purchase order for everything.

The bottom line question for these devices is: what type of response is required for what period of time? There are many shops where next day service is totally adequate except for month ends when they process the financials. In this case, it probably would be a good decision to have next-day service and have a reserve fund to pay the uplift should they need an immediate response during one of the 24 month-end days. In most cases like this they have already paid an uplift at some point in the past to get service between 5pm and 8am or on an HP holiday.

Another look at service needs is to separate the nonessential peripherals such as line printers and consoles from the four- hour response service for the system and the disc drives. In other words, if you have a critical need for printers, then have multiple printers rather than have the system printer on four- hour response. The typical difference between four hour response and next-day response is 20%, but you should get formal quotes before making any decisions.

CASE IN POINT:

John, a system manager for a large SNA print stations based on two HP3000 SERIES 42s each with an HP2680A laser printer. Everything was on standard four-hour response, the total annual hardware support cost was 45K. The systems were critical for two days following month-end processing on their IMB host, but for the rest of the month they could run on one system for one or two days if needed. It turned out that in the past year they had to pay for two uplifts for off-hours service. The solution was to place both printers on one of the systems on next-day coverage and budget in three uplifts per year. The end result was a savings of 9K per year.

As you go through this process you will find that a strategy will develop that is particular to your environment and business needs. Keep in mind that it is in the vendor's interest to drag out the quote and contract change process, and above all, to spread fear, uncertainty, and doubt.

System Software Contracts

Software support contracts are much easier to evaluate than their hardware counterparts. The first thing to consider is: do we use the level of support we have? If you have Account Management Support, when was the last time you saw or heard from your SE? If you have Response Center Support, when was the last time you called the response center?

The reason most sites purchase Account Management Support is because the system is new and this is the level of support the vendor recommends.

CASE IN POINT:

Jeff took over as MIS manager at a local school board. In reviewing the various support contracts for their systems, he discovered that they had an HP3000 series III on Account Management Support. When he inquired as to who the contact with the vendor was he found out that their SE hasn't worked for HP for two years and that the system hasn't been powered on for a year and a half.

In a meeting with HP Jeff tried to get a refund of the support charges but found that the school board had renewed the contracts, and there was no way for HP to know that the system wasn't in use.

Software support contracts have a main line item for the system support and have two types of other line items. The category support charge ends with +C00, and provides support for groups of software products such as languages, utilities, etc. The individual product line items end with +S00 and are charges for providing software and manual updates over and above the charge for the system.

For each of these items you need to decide if you are using this product now or will ever use it in the future.

CASE IN POINT:

Mary Jo, a system manager for a small health care company, reviews her software support contract and finds a category for office applications and a line item for hpdeskmanager. It turns out that they got the software as part of a promotion package when the series 70 was purchased and have no plans to ever implement electronic mail. In the past two years this free software package has cost them $3,120.

If you drop support for software products, then the standard HP line for restarting support is that you'll have to re-purchase the software product or pay the back months of support.

Applications Support Contracts

Most shops also have software support for some types of applications and utility software. The key things to evaluate are: are we using the application, and has the vendor been sending out regular updates to the product.

In many cases a utility package will be purchased and used for a particular purpose, then fall to disuse because of application or personnel changes. The rule of thumb should be if you aren't using the utility and don't plan to use it in the future, then you shouldn't be paying software support for it.

The other criteria for software support is new versions. If the vendor has not issued new versions or is not planning to issue new versions of the software, then paying for support is questionable.

One of the main values of participating in the various user groups is obtaining information on new hardware and software revisions. Another good source for this type of information is following the various trade publications such as THE

INTEREX PROCEEDINGS, INTERACT, and HP Professional.

AMERICAN DEVELOPMENT INC.

COMPUTER EQUIPMENT INVENTORY

Item	Description	Serial Number	Location/User	Ldev #	Cable Id
A1060AH	HP3000 SERIES 935	2839A11217	COMPUTER ROOM		
9133H	20MB DISC HP150	2512A13672	X463		
9133H	20MB DISC HP150	2512A14498	X462		
7980XC	1/2" TAPE DRIVE	2816A11323	COMPUTER ROOM	7	
7937XP	571 MB DISC	2816A20264	COMPUTER ROOM	4	
7937H	571 MB DISC	2709A02861	COMPUTER ROOM	1	
7937H	571 MB DISC	2709A30257	COMPUTER ROOM	2	
7937H	571 MB DISC	2709A30275	COMPUTER ROOM	3	
700/92	TERMINAL	2740A11806	X479	121	B09
700/92	TERMINAL	2740A11850	X480	120	B08
700/92	TERMINAL	2740A11924	X478	122	B10
700/92	TERMINAL	2740A11946	X477	123	B11
700/92	TERMINAL	2740A71945	X476	124	B12
700/92	TERMINAL	2815A11194	X475	125	C01
700/92	TERMINAL	2815A22265	X474	126	C02
700/92	TERMINAL	2815A22276	X473	127	C03
700/92	TERMINAL	2816A23610	X472	128	C04
700/92	TERMINAL	2824A16523	X470	130	C06
700/92	TERMINAL	2824A16584	X469	131	C07
700/92	TERMINAL	2824A26957	COMPUTER ROOM	100	
700/92	TERMINAL	2824A27570	X471	129	C05
700/92	TERMINAL	2838A05503	X468	132	C08
45979D	VECTRA ES	2812V08420	X451	148	D12
45979D	VECTRA ES	2812V19152	X452	147	D11
45979D	VECTRA ES	2812V25182	X450	149	E01
45945D	VECTRA ES/12	2825A25312	X442	157	E09
45945D	VECTRA ES/12	2825A25424	X447	152	E04
45945D	VECTRA ES/12	2825A25427	X446	153	E05
45945D	VECTRA ES/12	2825A25429	X445	154	E06
45945D	VECTRA ES/12	2825A25431	X444	155	E07
45945D	VECTRA ES/12	2825A25434	X443	156	E08
45945D	VECTRA ES/12	2825A25438	X441	158	E10
45945D	VECTRA ES/12	2825A25466	X449	150	E02
45945D	VECTRA ES/12	2825A25474	X440	159	E11
45945D	VECTRA ES/12	2825A25642	X439	160	E12
45945D	VECTRA ES/12	2825A28419	X448	151	E03
45945A	VECTRA MODLE 45	2741A56666	X464	135	C11
45940A	VECTRA MODLE 59	2716A13421	X458	141	D05
45940A	VECTRA MODLE 59	2716A13640	X455	144	D08
45940A	VECTRA MODLE 59	2716A16336	X454	145	D09
45940A	VECTRA MODLE 59	2716A21027	X453	146	D10
45940A	VECTRA MODLE 59	2716A24532	X457	142	D06
45940A	VECTRA MODLE 59	2716A24576	X456	143	D07
45935A	VECTRA MODLE 35	2505A11548	X466	134	C10
45935A	VECTRA MODLE 35	2505A11549	X465		
45935A	VECTRA MODLE 35	2505A11554	X467	133	C09

AMERICAN DEVELOPMENT INC.
COMPUTER EQUIPMENT INVENTORY

Item	Description	Serial Number	Location/User	Ldev #	Cable Id
45849A	HP150 TOUCHSCREEN	2521A17271	X463	136	C12
45849A	HP150 TOUCHSCREEN	2521A17271	X461	138	D02
45849A	HP150 TOUCHSCREEN	2521A17353	X460	139	D03
45849A	HP150 TOUCHSCREEN	2521A17510	X462	137	D01
45849A	HP150 TOUCHSCREEN	2521A19984	X459	140	D04
45816A	20MB DISC VECTRA	2552A33093	X466		
45816A	20MB DISC VECTRA	2552A33094	X465		
45816A	20MB DISC VECTRA	2552A34034	X467		
3630A	COLOR PRO PRINTER	2728A51905	X454		
33440A	SERIES II PRINTER	2652J12674	X467		
33440A	SERIES II PRINTER	2652J29233	X466		
33440A	SERIES II PRINTER	2805A41849	X465		
2686D	LASER JET PRINTER	2604J04215	X455		
2686D	LASER JET PRINTER	2604J04248	X454		
2686D	LASER JET PRINTER	2604J04248	X458		
2686D	LASER JET PRINTER	2604J04289	X456		
2686D	LASER JET PRINTER	2604J04402	X457		
2628A	TERMINAL	2405V13424	X498	102	A02
2628A	TERMINAL	2405V65847	X497	103	A03
2628A	TERMINAL	2426V56842	X496	104	A04
2628A	TERMINAL	2429V56497	X495	105	A05
2622A	TERMINAL	2342A83123	X499	101	A01
2602A	PRINTER	2419A11706	X462		
2601A	PRINTER	2427A14244	X463		
2564B	600 LPM PRINTER	2836A16946	COMPUTER ROOM	5	
2564B	600 LPM PRINTER	2836A26957	COMPUTER ROOM	6	
2563A	300 LPM PRINTER	2428A14077	COMPUTER ROOM		
2392A	TERMINAL	2603A36148	X494	106	A06
2392A	TERMINAL	2603A94018	X488	112	A12
2392A	TERMINAL	2603A94027	X493	107	A07
2392A	TERMINAL	2603A94028	X491	109	A09
2392A	TERMINAL	2603A94029	X490	110	A10
2392A	TERMINAL	2603A94033	X483	117	B05
2392A	TERMINAL	2603A94058	X486	114	B02
2392A	TERMINAL	2603A94067	X484	116	B04
2392A	TERMINAL	2603A94069	X481	119	B07
2392A	TERMINAL	2603A94072	X482	118	B06
2392A	TERMINAL	2603A94893	X492	108	A08
2392A	TERMINAL	2603A94903	X487	113	B01
2392A	TERMINAL	2603A94904	X485	115	B03
2392A	TERMINAL	2603A94999	X489	111	A11
2345A	DIST TERMINAL CTL	2813A12573	COMPUTER ROOM		
2345A	DIST TERMINAL CTL	2813A12574	COMPUTER ROOM		

The Perils of Writing a Security Policy

Jim Knight
MEDSTAT Systems, Inc.
777 East Eisenhower Parkway
Ann Arbor, MI 48108
(313) 996-1180

Overview

Developing a security policy can be a very frustrating experience. No one will help you because the information is "Confidential". So how do you know what belongs in your security policy, or how it should be designed? This paper will detail the efforts required in writing a security policy, an how to avoid the pitfalls therein. What is a security policy? What does a security policy look like? Why won't anyone show me their security policy? Where can I get more information? How do I decide what needs to be included in my security policy? All these questions and more will be answered within the realm of how we have developed and are developing our security program.

What is a Security Policy?

A security policy is just one part of an overall security plan. It is the foundation for all other security standards and policies within the security plan. It should be a statement that indicates what value the company places on information and security. As a short document of 1-5 pages, it should be precise in stating not only the value of the data as a company asset, but also in its exact expectations of users. It should also clearly state that the company will take action against those who misuse company data or assets. It should be understood that a security policy is not a detailed procedure, but instead a directive from management.

What does a Security Policy look like?

When I first set out to write a security policy, this was the most frustrating part. I had no idea of where I should start, or what the format should be like. I was unsure whether it is should be very detailed, or more global in orientation. After some research and some help from other sources that I will cover later I came to the determination that at least the following points should be covered:

- Statement of Purpose
- Scope
- Classification of Users
- Notice of Intent to Enforce

The Statement of Purpose is the heart of the security policy. It should recognize the need, and the intent to protect the assets of the company. It should also define those things that are to be considered covered by the security policy. For example, data, software, procedures, processes, and computing resources are things that might be considered assets to be protected by the policy. This policy should be agreed on by the company management, and be in line with the general policies and strategies of the company.

The Scope statement should define just what is the realm of responsibility that the Security policy is meant to cover. It should state which departments, branches, or classes of users/employees fall under the reign of the policy. It may also describe the goals of the security policy. In general these goals are prevention, protection, detection, damage assessment and recovery. It should also define which assets are to be considered within the protection of the policy.

The Classification of Users is used to delineate lines of responsibilities between users. Certain users are by function going to handle more sensitive data than others. In general, this part of the security policy is used to define the role of the Security Administrator. Other areas that might be defined are owners, management, users, clients and contractors. Some definition of these classes should be given along with the type of asset that they may be involved with.

The Notice of Intent to Enforce the security policy should be a statement that explains the company's willingness to discipline offenders if the standards and policies of the company are not upheld. It should include wording to the effect that if the standards of the security policy are broken, the Company will punish the offender by whatever means necessary including but not limited to termination of employment. It should be a strong stance, leaving no doubt that the company is serious about protecting its assets.

Depending on your company, you may want to consider some additional points. These may be required by the company management, the security committee, or others who play a part in the development of the policy. Since this may be considered a legal document, it may be good to have your company's lawyers read and approve it. They may also have some recommendations for items that should

be included in the policy. The main point to remember is that there are no hard and fast rules to what a security policy can or cannot contain.

Some other items of interest that might also be covered:

- Frequency of Review and Update
- Audit functions expected
- Enforcement Responsibility
- Authorization of Exceptions

The Frequency of Review and update should be somewhat open ended. There should be a maximum allowable time between updates, but not a minimum. Probably as a minimum the security policy should be reviewed and updated once a year. Remember that as your business and company changes, so will the things that you are trying to protect. An out of date security policy quickly becomes useless.

Audit functions that are expected should be detailed as part of the responsibilities of the Security Manager. They might be included here to show that you are actively checking for breeches of the security policy. Image is probably one of the most important parts of the security policy. It the security policy comes across as just something to satisfy management, and not something that will be actively enforced, then it will be useless.

Enforcement Responsibility often falls on the Security Manager, however they may be delegated to members of Human Resources, upper management, or a security review committee. It may be important to detail who will take responsibility for breeches of the Security Policy. If so, this may be an important part of your policy.

Although the Security Policy should be set in stone, there may be incidents where exceptions are necessary. If such a need should be expected, then procedures to handle such occasion should be detailed here. If at all possible, it might be best to leave the Security Policy as a stand alone document, and allow for exceptions by overlapping the responsibilities of different user classes defined in the user class responsibilities.

Why won't anyone show me their security policy?

The Security Policy will eventually detail the things that are most critical to your company. If a company or an individual of that company were to release their security policy to outside individuals, they would be releasing the points where the company is most vulnerable to sabotage. This is what makes it important to keep the Security Policy, and the detailed class responsibilities list, strictly confidential. It is probably also in the best interests of the company that confidentiality agreement be presented with the security policy and signed by the employees.

Where Can I get more information?

This question was the biggest question I had when I left the 1990 Boston Interex Conference. My principle goal when I went to the Interex Conference was to find out what I should do in writing my security policy. In this respect, at first I felt like the conference had been a failure for me. However, I then realized that the real answer is that there is information available for the asking.

The first place to ask is with the vendors of Security products for the HP/3000. If you are already using a particular security package, I would suggest you get with a representative of that company and discuss your concerns for information on writing a security policy. If you are not currently using a security package, use that as a bargaining point with the software vendors. A good helpful vendor will be able to supply you with some information that will make your life a lot easier.

The Second place you can go is to the Security Roundtable and/or other security presentations like this one. Be sure to ask questions of the speakers, and whenever possible make acquaintances with others in attendance. As with most other topics discussed at the conference, the networking between your peer at the conference becomes a valuable resource.

The Third place to investigate would be the Computer Security Institute. This group publishes a Computer Security Handbook which you should find invaluable in designing your security policy. It contains some sample security policies along with a wealth of information on developing a security plan.

Last but not least, your company's auditors may be a good source of information. They should be familiar with security standards, and probably will be able to help you design a policy that closely matches the needs of your company.

How do I decide what should be included in the Policy?

The first step to deciding what should be included is to determine what your assets and risks are. A good first step towards this end would be to have an outside party come in and do a security review and/or risk analysis. This process will point out the things that are most important to your company as well as to show your weakest areas of security. By taking these points and analyzing them closely you should be able to form a security policy that will fit the needs of your company.

Security Policies in the HP/3000 environment.

The HP/3000 Environment allows for a good breakdown in user responsibilities. Because of the way HP security is implemented, several distinct user classes can be defined with specific realms of responsibility. In general these classes are broken down by the following capabilities:

- SM System Manager
- OP System Operator
- PM Privileged User
- AM Account Manager
- AL Account Librarian
- GL Group Librarian

The SM or Systems Manager class of users are the users with the most responsibilities. They are responsible for every user, account, group and file on the system. This group should consist of a small group of users preferably less than five.

The OP or System Operator class of users are responsible for every file on the system. They have the capability to store and restore every file on the system. This alone makes them an important part of the security structure.

The PM or Privileged Mode class of users have complete control of the system. Along with the SM class of users, they have complete control of the system. Ideally this class should have as few users as possible if any.

The AM or Account Manager class of users have control over all users, groups, and files within a specific account. Account managers should have responsibilities as required to protect the resources owned or used by that account.

The AL or Account Librarian, and GL or Group Librarian are classes which are very similar in structure. They give control of the files of a specific account, or a specific group in an account to a user. These capabilities can be used to further divide the responsibilities of users below the account manager level.

The HP environment is ideal for spreading responsibilities of users for their assets over a wide range of users. It allows a few trusted users to have the greatest level of responsibility, while allowing lots of users to take on smaller areas of responsibility. While this structure may not necessarily fit into your environment, it can give you a good idea of the possible classes of users that you may have in your environment.

So what are the perils in writing a security policy?

The perils in writing a security policy are many. Of the most difficult that I have had in my effort to write a security policy, the following stick out:

- What do I put in my security policy.
- Who should write the security policy and have a part in the decision making process.
- What kind of stance should the security policy take.

I have to say that the most difficult obstacle I ran into in trying to develop a security policy was to find out what belonged in the security policy itself, and in what kind of format it should be expressed. This in itself is what drove me to write this paper. I saw the

distinct need to pass on to others the ideas and format of what a security policy should contain. It should be noted that while the ideas here are basically sound, they are mine, and might not be perfect for any given company. However, this should at least ease the burden that is being placed on an individual or group who are out to write a security policy.

As for who should write the security policy and have a part in the decision making process, I have found that many have misconceptions about this. When I was assigned the duty of writing a security policy, I thought it was something I would be able to go off into a corner and write, and then present to the users. Instead, I found out that this should really be a company standard, with the blessing of management. If this policy is just a document that I created without the full backing and knowledge of the upper management, it would be useless. It is important that the policy not only be embraced by the company as a whole, but also enforced. If a situation arises where a breach of the security policy occurs and there is no discipline handed out, the security policy basically becomes null and void. This is why it is important for the policy to garner top down support.

As for the stance that the security policy should take, I soon realized that there were several approaches that could be taken. A general statement could be issued about security standards, a policy could be issued about security standards, or a policy could be issued with the documentation of responsibilities and standards necessary included along with it. In this day and age I find it extremely critical that a security policy be definitive in its stance. A weak security policy will lead to weak procedures and eventually to no security policy at all. On the other hand, a strong security policy will protect the company and its employees for the years to come.

As part of the process of writing the security policy, detailed user class responsibilities should be developed. These should ride along with the Security Policy to give specific guidelines to each employee. In this way, the more sensitive responsibilities can be documented only to the most important employees, and the most general responsibilities can fall on the shoulders of all employees. As was demonstrated in the explanation of sample classes for the HP/3000 systems, A users responsibility should be determined by his ability to control assets of the company.

Finally, a security policy is only the foundation of a good overall security plan. With it in place, a security plan can be built on a good foundation, and company assets can be protected in the most secure way possible. However, the security policy is just the first step towards a successful security plan.

John Doe Corporation
Security Policy
Effective Date: January 1, 1995

Purpose:

The John Doe Corporation takes pride in the business it has established. As such, certain areas of the company have become more and more important. All data, cpu resources, and software are vital to the continuing operation of the company. This document is to set forth the standards necessary for the protection of these assets.

Scope:

This policy shall apply to all employees and establishments of the John Doe Company both present and future. It shall apply to all data, cpu resources and software associated with the company. This shall include but not be limited to data created and/or maintained by the John Doe Company.

Responsibility:

The Security Manager will be responsible for reviewing, updating and enforcing all aspects of this policy.

The Systems Management will be responsible for maintaining all accounts and files in a secure manner across all system boundaries.

The System Users will be responsible for maintaining all files within their own areas in a secure manner. They will also be for all passwords and accounts under their control.

Enforcement:

It is the desire of the John Doe company to strive to protect it's assets. Any violations of this policy after it has been instituted will be presented to company management for appropriate action, including but not limited to termination of employment

HIGH WATER AND BROKEN BRIDGES

GAYLORD MAINES

BAHLSEN INC.
ONE QUALITY LANE
CARY, NC 27513
(919)677-3260

In October of 1990, the barrier islands off the North Carolina coast experienced the worst flooding in over 30 years. Contingency plans had been developed and consistently updated for the evacuation of Hatteras and Ocracoke Islands. Every major plan depended on the single bridge that offered evacuation, electricity, phone service and datacommunications. The storm that brought the high water also freed a dredge from it's moorings. The link to the mainland was severed when the dredge impacted on the bridge. As 300 feet of bridge fell into the Atlantic, years of planning went with it.

Under all existing contingency plans, evacuation of Hatteras and Ocracoke Island was to take place over the Bonner Bridge. Residents of Ocracoke Island were to take the ferry to Hatteras and drive north to the bridge. Nobody had thought about what to do in the event that the bridge was not accessible.

Information vs Misinformation

Disaster Recovery planning is the process of being prepared for a crisis or disaster. This is more an art than a science when actual implementation begins. Natural or manmade, catastrophic events rarely follow a path that was predetermined by ourselves. The planning process allows us to accommodate as many "If/Then" situations as we can imagine. But survival hangs in the balance of our actions. These actions are driven by decisions that are accelerated, if not spontaneous. Our decisions, hopefully, are based on information, though sometimes misinformation.

In the early hours of a disaster, it must be assumed that a high percentage of information received is in error if not completely false. During this period the source of the information rarely adds value to it's accuracy. The difficult task is to determine what is true and what is not true.

With this in mind, it becomes clear that we must surround ourselves with as many sources for information as possible. Obviously, these sources will serve better if they suffer little to no effect from the events that put us in survival and recovery mode. Some of the following sources will allow communications as well as information gathering.

- Citizens Band Radio (CB)
- Ham Radio
- Marine Band Radio
- Police Band Radio
- AM/FM Radio
- Cellular Phones
- Portable TV
- Word of Mouth

It should be remembered, however, that all of these should have power sources that are independent of building power. Just assume that power from the local electric company will be out for a period of time. All of these should be powered by:

- Batteries
- Automobile Electric
- Generator
- Solar

There are even 1 meter satellite dishes available now. They pack in two suitcases and are quite mobile. They will allow you to tap directly into the international AT&T network.

Misinformation can have devastating effects. In the case of the Hatteras storm, misinformation caused a disastrous evacuation of the island that took days.

Word spread, at least unofficially, that evacuation would be a 45 minute ferry ride to Ocracoke, drive across the island and then a 2½ hour ferry to the mainland. People began lining up at the Hatteras ferry. Many non-residents had no clue as to where they were going. Some had no idea that they were leaving one island, only to go to another. So they lined up for miles and waited. Many gave no thought to provisions, even though many stores were open and running on generators. After waiting 12-18 hours, they finally found themselves bound for Ocracoke via a state owned and operated ferry. Weary and unaware, they left the frying pan for the fire.

Two things should be made clear at this point. First, Ocracoke is an island about 14 miles long with a village at the southern end. The population is about 500. With the exception of this village, there is virtually nothing else on the island. No houses, no stores, and no shelter. As people got off the ferry on the northern end of the island they had time to appreciate the desolation as they made their way towards the next ferry crossing. But the line of cars soon exceeded 10 miles. Needless to say, they were given ample time to inspect their new surroundings.

The other thing is to understand that the ferries are run by the North Carolina Highway Department. They displayed the

ability to get people to Ocracoke at about four times the speed that they could port them to the mainland. The island began to suffer the only traffic jam since it was discovered 400 years ago.

Residents became concerned for the health and well being of all concerned. Word went out, again unofficially, to stop sending people over. But, since nobody appeared to truly be in charge, this was quickly and completely ignored. The Hatteras to Ocracoke ferries ran 24 hours a day.

Ultimately, evacuees settled in for a 36 to 48 hour wait. The military flew in sleeping bags and distributed them while supplies lasted. A few portable restrooms were scattered along the road. A Red Cross truck arrived from the mainland, totally unprepared for the number of people stranded, with 60 cups of coffee and 2 dozen donuts. The presence of the military helped keep the peace as fighting broke out over the few provisions available.

The residents of the island set up a kitchen and shelter for these poor souls. Unfortunately, they forgot to tell many people about it. Those who knew were afraid to leave their vehicles unattended.

The results of misinformation can, in fact, be devastating. The unfortunate results in this case are endless. What's significant to remember is that negative results of misinformation will compound themselves rapidly.

Human Behavior and It's Unpredictability

I think we'll all agree that we fear what we don't know. In a disaster, very few things are known. Fear and confusion spread immediately. Human behavior can run to extremes; for good and for bad.

Fear can quickly lead to panic. In this event, many people immediately got into their vehicles and began to drive, but they hadn't a clue as to where they would go. A single highway runs north to south on this island. A broken bridge was on one end, and ferries were unable to run on the other end. Still, people were on their way.

Several businesses were open and running on generator power. Naturally, a run on provisions took place. Many logical items were cleaned out in dry goods. Lanterns, fuel, flashlights and the like were gone quickly.

What was more interesting was the types of food that people bought. In a time of no electricity or refrigeration, milk, produce, and meats were immediately sold out. It was my observation that bottled water and canned goods would have held out for as long it would have taken me to consume it all myself. Snack foods and soft drinks sold well also.

So here's the big picture. Hundreds of vehicles wandering up and down a single highway with no where to go. Families, fishermen, residents, and surfers were wet, wind-blown and panicked. The road was, in places, under 3 feet of saltwater. Gasoline wasn't available. Cars began stalling. The water continued to rise.

Hours turned into days. It became clear that cornchips, raw hamburger, and warm milk were not in fact the bare necessities. The milk of human kindness was spoiling due to the unavailability of ice. Civil disobedience broke loose over restrooms. Generally speaking, a gloomy mood prevailed over many parts of the island.

What's interesting in this is the change in the weather conditions. Near the end of the 1st day, the storm began to break. The following day was clear and sunny and the wind began to ease off. The 3rd day was better still. The water had not only receded, but most of the damage it had caused was not immediately visible. Still the panic to evacuate was running with a full head of steam. Telephone service had been restored in some areas but this seemed to generate more anxiety as people clamored to call the mainland.

Human behavior in a disaster is predictable to the degree that it will become radical to some degree. The more that people are left to their own devices, the more radical they will become.

People need advice and information. Without it, they tend to buy perishable products and spend their time driving through saltwater flood zones nurturing ugly moods.

Successes/Failures

Keeping in mind that the area in question is a resort and vacation spot, most businesses are retail. This storm hit in the middle of a peak business season and all but isolated the islands for months to come. Only through preparation and adaptability did many businesses survive.

One thing that seemed to be an indicator to the long term survival of a business was the ability to stay open during the days that immediately followed the storm and the collapse of the bridge. This gave a critical, psychological advantage to business owners and employees. Their customers, whether resident or transient, were left with a positive impression. Existing relationships were strengthened and new ones were established. Ad hoc business agreements were arranged. The barter system played a significant role since business continuation was the primary concern. Sharing fuel supplies in exchange for shared transportation is an example of the mutual benefits in these arrangements.

One fundamental requirement was self sufficiency in electrical power. Without a generator and large supply of fuel, business continuation is virtually impossible. The generator needs to be all but capable of supplying 100% of the electric requirements under normal conditions. Most aspects of a business seem to be critical functions when threatened with discontinuation. This is true even when the interruption is temporary. Even those areas and tasks that can be suspended are balanced by the energy required for a new but quite necessary need brought about by the crisis at hand.

In a computer dependent business, electrical power is obviously a must. What's equally important is the distribution and isolation of these circuits. Switching power supplies can be dramatic enough to sensitive hardware without introducing overloaded electrical circuits.

Since datacommunications and phones were down, decisions were required. Could any given organization continue to process without a datacomm link to the mainland? Obviously the answer varied from business to business.

Batch processing environments suffered the least. An example would be a retailer that dumps to a mainframe nightly from point-of-sale devices. The primary concern is routing data collection so that posting can be completed at whatever time the link is established. The objective here is a thoroughly tested collection process with sufficient capacity.

Those businesses that rely heavily on on-line access to other systems were in a much more difficult position. Without a contingency plan for on-line processing, the choice was simple. Go to manual systems or close the doors. Either way the risk and exposure level increased dramatically.

Local bank branches simply closed their doors. Their losses were acceptable if not minimized. Retailers and service organizations were not so fortunate. The days that immediately followed were the last opportunity at revenue for months to come. In retrospect, the money exchanged during that short period in time made an extremely critical difference for those businesses that remained open. The losses simply buried some of the others.

The Basic Necessities of Disaster Recovery

The first and last requirement of Disaster Recovery is a plan. Without forethought and planning, you can only react in the event of a disaster. The conditions will be set for failure.

The most important element in surviving a disaster is people. Your associates must be present and collected first and foremost. They must be at ease about the safety and security of their families. Any property damage that they may have

experienced must be addressed. It is the responsibility of the organization to offer as much assistance as possible or accept the consequences of human reactions. Frequent and unannounced absences will occur. Depression is likely in the days that follow. Resignations are not uncommon due to the emotional condition of your associates. Psychological and financial assistance must be made readily available to maintain the stability of your organization.

In further consideration of your associates, remember the basics required for survival, food, shelter and water are undeniable. A warm place to sleep and emergency lighting. Childcare and medical assistance are necessary. Cash is essential since banks may be closed and credit cards will be useless. Without electricity, gasoline will be unavailable.

Remember that people will not always recognize the seriousness of their positions in a logical manner. Their reactions may well reflect an acknowledgement, but their decisions may display denial. This is why they arm themselves with cornchips rather than canned goods. But, if your organization anticipates that its' associates will be there from disaster survival though recovery, then it is the responsibility of the organization to recognize and compensate for these tendencies.

Information is another necessity. Inbound and outbound. Every form of available communications must be considered and implemented. This can range from cellular phone to CNN broadcasts. Pre-designated out of state contacts should be utilized. A single individual, located outside of a disaster area, can accommodate communication needs ranging from vendor contacts to family members.

An emergency information center should be established for the benefit of all. A spokesperson or persons, should be assigned the responsibility of disseminating information and clarifying misinformation. These individuals must have the authority to announce areas requiring action and know that they will be tended.

Liaisons must be established with local authorities. The police and fire departments will be much more receptive to a single source of authority representing your organization, than to random contacts from multiple individuals.

A Disaster Recovery team should be organized in advance. Per your plan, each of these players will have their own area of responsibility. Cross-training or double coverage should be considered to insure that all critical areas are covered in the event that one or more players are unavailable. This team should be easily recognized by all concerned. This can be accomplished with a specific hat, shirt, or a sign in the car window.

And again, you need a plan. As plan development occurs within your organization, more and more areas of concern will present themselves. Each must be weighed and classified as critical or non-critical. There is no amount of planning that should be considered excessive unless it becomes too complicated to be practical or it exceeds, in costs, the valve of the business.

How to Commit (or Prevent) Computer Crime

Louis R. Mills, CDP

Bio-Rad Laboratories, Inc.
1000 Alfred Nobel Drive
Hercules, CA 94547

(415) 724-5613

Do you like money? Would you like to make $50,000 tax free this year? Is a million dollars in a Swiss bank account worth a few hours work each week? Then perhaps you need to consider a career change to Computer Crime.

Consider this quote by Malcolm Senn, taken from a 1984 Omni Magazine article by Roger Rapoport:

> If I had been a traditional embezzler, they probably would have arrested me within six months. But with the computer, I never had to touch the cashbox. Being the Controller made it easy. First, I set up fourteen phony suppliers. Then I programmed the unit to pay automatically for nonexistent goods and services from my dummy companies. That way, I could be skiing in the Alps while my California employer mailed checks to these fronts.
>
> The system earned me a million dollars over six years. By then, I was ready to quit and start enjoying the houses, planes and boats I'd accumulated. I knew if I just left, my successor eventually would figure out what I'd done. Since I can't stand suspense, I began leaving clues to help the auditors catch me. I wanted my day in court, serve a brief sentence at a country-club prison, and live off of the money I'd stashed in Switzerland.
>
> But the auditors were impossible slow. Finally, in desperation, I started bouncing checks. It still took them another three months to catch up with me.

Malcolm was caught, tried and convicted. To his surprise, he spent five years at San Quentin prison before being paroled. He did not return the money. While in prison, he was the warden's assistant. He arranged the installation of a computer terminal so that inmates could take a computer programming class. Many of his students are now out, in society. Is one working in your firm?

Malcolm had no criminal record, had never stolen before, and was a trusted and valued employee. In fact, he fit the profile for computer crime perfectly.

The Federal Bureau of Investigation (FBI) uses a broad definition of computer crime:

Any crime where the computer is either:

A. The vehicle to commit a crime
B. The victim of a criminal act

Why choose a career in computer crime? Why not just rob banks for a living? Well, according to the FBI, the average bank robbery (1987):

1. Nets $6,325
2. Is quick. Under two minutes spent in the bank.
3. Dangerous! Armed guards and police.
4. Lots of physical evidence (Smile!) and witnesses.
5. 100% of these crimes are reported. (It's the law.)
6. 60% conviction rate, with few suspended sentences.

Now a computer criminal, on the other hand, has these FBI statistics going for him. The average computer fraud:

1. Nets $50,000+ (The FBI only handles $100,000 up.)
2. Takes 1 to 5 weeks.
3. Virtually no danger.
4. Little or no evidence -- you can erase it yourself.
5. 1% to 10% reported -- FBI estimates 2%.
6. 1% to 10% convicted, often for a lesser but included offense. Suspended sentences not uncommon.

So a little math tells you that if only 2% are reported, even if 10% of those are convicted, you only have one chance in 500 of either doing time or paying a fine. Not bad odds.

But good managers set high goals for themselves. Let's set some goals for you. Let's take a 1987 example from Volkswagen in Europe:

1. Net of approx. $259,000,000.
2. Took less than one year.
3. No one was physically harmed.
4. Little evidence was found.
5. One insider charged, seven outsiders indicted.
6. One conviction so far.

But some of you are real movers and shakers. $259 million is small potatoes. Let's set our sights on some real money. What has been the biggest scam so far? Equity Funding Insurance:

1. Fraud amount $2,100,000,000.

2. Built up over three years.

3. No danger.

4. Less than 10% of fraudulent entries found.

5. Many executives and workers charged with fraud.

6. 13 convictions of senior executives in federal court under the Foreign Corrupt Practices Act.

How do people get away with crimes like these? Won't other people turn you in? The reality is society's attitude about ethics. Ethics are often "situational". And if people don't perceive a direct harm, they often ignore criminal acts. It was not uncommon 10 years ago for graduate classes in Computer Science to give credit to students who could violate the security on the school mainframe. When a professor was asked about this at a National Computer Conference, his reply was:

"We're not here to teach ethics; they should have learned them before they arrived on campus."

Most of us in the computer industry often wonder about the ethics of vendors' marketing claims, pre-release announcements, the credibility of consultants and repairmen, etc. There has been a surge of interest in ethics in American business and MBA programs since the "insider trading" scandals of the last few years, so we may see a renewed interest in this area. The Summer 1989 issue of DPMA's Information Executive carried four articles on ethics.

But a typical computer crime is a conspiracy, and this case is a good example of two ethical failures:

In Chicago, a company president, the DP manager and a programmer were charged with a $40,000,000 fraud for inflating the inventory records to cover up poor management. The company board of directors decided to slowly deflate the figures to avoid a drop in the firm's stock price.

Both the Securities and Exchange Commission (SEC) and the Justice Department said "NO!"

The good news is that computer crime seldom closes down a business. Dun and Bradstreet provided these figures in 1987 for the reasons businesses fail:

0.3%	Fraud
0.6%	Neglect
1.3%	Disaster
9.8%	Lack of management experience
9.9%	Lack of experience in the line or industry
23.8%	Unbalanced experience
54.3%	Incompetence

Most of you know someone in upper management who fits into one of those last two categories.

But even more interesting to me are these figures from the US Justice Department:

80% of frauds start from unintentional errors.

15% start from "What if ..." scenarios.

5% are purposeful acts.

Computer Crime can be defined in three different categories and we will review two case studies in each to give you a feel for the areas you may want to pursue.

(Case study information is not published since it deals with techniques used to commit a criminal act. The publishing of such information is a crime in many places.)

1. Physical access crimes
   - A. Impersonation
   - B. One-dimensional security penetration
2. Transactional crimes
   - A. Financial fraud, embezzlement, etc.
   - B. Theft of goods or services
3. Electronic (programming) acts
   - A. "Salami swindles", etc.
   - B. Sysop or "techie" techniques

Computer crimes, like many White-Collar Crimes, have a low reporting rate. Remember the FBI estimate of two percent? Why?

1. Embarrassment felt by upper management.
2. The impact of loss of public trust in the company.
3. The potential effect on the company stock price.
4. Fraud suspected, but not provable or detected.
5. Culprits are unknown.
6. Legal difficulties from:
   A. Lack of evidence
   B. Conflicting laws or enforcement agencies
   C. Technological ignorance of management, police, judges, juries, etc.
   D. Jury sympathies
   E. Seizure of business records, disks, computers...

If you're in a company that is the victim of a computer crime, what is the recommended order of bringing charges?

1. Foreign Corrupt Practices Act
2. Federal/state fraud or embezzlement statutes.
3. Theft of goods or services.
4. Vandalism or sabotage.
5. Trespassing
6. Computer crime laws.

In the US, there is the Federal Computer Security Act of 1987 that makes you liable if you were involved in the design, implementation or use of a "dis-approved" computer application. So, if the system is certified for security, it isn't your fault if you broke into it, is it? Have them bring charges against the guy who designed it.

Don't **fire** or demote someone who is suspected of a criminal act. In many states, your action is illegal. You must have a **conviction** before you can act.

The following **management recommendations** are common policy in many organizations:

1. Keep DP staff off of production systems.
2. Identify employees with harm potential.
3. Restrict physical access to media & equipment.
4. Require pre-employment screening.
5. Vacations must be taken.
6. Rotate job assignments.
7. Use decoy names in address files.
8. Don't let people work alone.
9. Build in audit trails and controls.
10. Use exception reporting, highlight unusual activity.
11. Be ready to PROSECUTE!

Here are some specific **Security suggestions.**

1. Avoid high risk people in key jobs.
2. "Fair and Equal" treatment must be the perception.
3. Educate and drill on security.
4. Manage by walking around & talking.
5. Track severe personal problems without violating an employee's privacy.
6. Shred or burn your own trash. Don't let it sit out.
7. Require and verify proper ID.
8. Verify employment and education claims.
9. Know the local gendarmes and authorities.
10. Know what constitutes "best evidence".

There is a wealth of material about computer security and computer crime. I recommend any book by the following:

Donn B. Parker, Stanford Research Institute, CA

August Bequai, Attorney, Washington, DC

Jack Bologna, Computer Protection Systems, Plymouth, MI

Additionally, the following are excellent sources:

1. Computer Crime Legislative Resource Manual, US Dept. Justice, Bureau of Justice Statistics. 150 pages, price unknown. Read this first, before you take a suspect case to the authorities.

2. Guideline for Computer Security Certification and Accreditation. NBS FIPS-PUB-102, 95 pages. Nat'l Technical Info. Service, Springfield, VA 22161. $11.50

3. Computer Security Digest, CPS, Inc. 150 North Main St., Plymouth, MI 48170. $110 for 12 monthly issues.

For the current status of state or federal regulations:

Joseph Collins
DPMA Headquarters
505 Busse Highway
Park Ridge, IL 60068-3191

The Information Terrorist -
The Computer Virus

Louis R. Mills, CDP

Bio-Rad Laboratories, Inc.
1000 Alfred Nobel Drive
Hercules, CA 94547

(415) 724-5613

In 1977, Thomas J. Ryan wrote "The Adolescence of P-1". It was the story of a program that moved from computer to computer, taking control of the Operating System of 7,000 computers. It was fiction. Yet the Robert Morris Worm, a program written by a Cornell University graduate student only 11 years later, is reality. The program was designed to hide itself and defend against removal. It could adapt itself to at least two radically different computer architectures. It had over 50,000 lines of code and one major bug -- it replicated over 100 times faster than planned. It was written as a worm with a "trap door" feature to capture and collect E-mail addresses and passwords, with the virus-like capacity to store itself on each new system it encountered. But it never properly detected that it had stored itself, so it kept on copying itself!

It infected 6,200 mainframe computers on the world's largest computer network and had most of the systems down for two days, and some of the nodes were down for a week.

Why the title for this paper? Because when I started collecting information on viruses for a talk to a local chapter of the Assoc. of Records Managers and Administrators, there were about 30 known viruses for the IBM/PC and almost as many for the Macintosh/Apple family. That was in February of 1989.

Today there are over 100 viruses for these two families of personal computers. And a virus is not a piece of software to challenge the intellect of the person who wrote it. It is a deliberate attempt to destroy or limit the usefulness of someone else's computer. The victims are often innocent users, unknown to the perpetrator. And like a terrorist putting a bomb on a plane, the author of a virus doesn't care who gets hurt and certainly doesn't want to take credit for the crime. It is an ethical and moral outrage.

And because of the nature of a computer virus, often many more people can be impacted than expected. The economic loss is often hard to measure compared with the direct cost in time and effort spent in recovering. Worse yet, according to Datamation's 1989 survey, nine out of ten installations that get infected experience a relapse within a week of cleaning out a virus.

So let's define some terms before we go any further.

Bootstrap - a small number of machine instructions found in a particular location on disk, diskette or ROM. Mainframers often call this the "IPL code". These instructions tell the computer where to find the Operating System.

Time bomb - a piece of code that is triggered either by a date or number of times executed. The Columbus Day virus attacks your computer's hard drive the first (or tenth) time you boot it up after Columbus Day.

Trap door - any software that performs a function other than the obvious one. For example, in a utility that has user exits, you could provide logic to copy user data to a specific file on another account for later display and analysis.

Trojan Horse - a "logic bomb" hidden in some piece of software that looks very innocent. It may or may not be a virus. Usually designed to be a "one time" program.

Worm - a program that consumes system resources to perform a task. A worm may search parts of a system looking for passwords every time someone calls the COBOL compiler. These are usually complex programs that turn themselves off after a short period so that they aren't noticed. But some worms can be destructive. The Morris Worm was a worm program with virus capabilities.

Virus - a virus is any piece of code that attempts to replicate itself by attaching (infecting) the same or similar code onto other systems or disks.

So, everyone's heard of the not-so-mythical programmer who buried code in the payroll system to wipe out the employee payroll records if the program discovered that he wasn't being paid that month. Well, one man's name was Burleson, and he wiped 168,000 records from a payroll database only two days after he was fired. His "time bomb" held up paychecks for an insurance company for more than a month. He was convicted of computer sabotage under a three-year-old Texas law in 1988.

A Trojan Horse can be a combination of things, sometimes benign, that may look innocent -- like a game or an idle display on the Mac -- or an enhanced directory program --that hides a Time Bomb, Worm or Viral replicator. It does what you want and expect it to do, until it is triggered, either by an event or by a change in environment. For the latter, it may mean that the program works fine when from diskette, but detects when it has been loaded from the C: drive and then corrupts your directory of files. Or it may simply wait for the first time you run it on the first day of the fourth month, then display an April Fool message before giving you back its normal operation.

But a virus has a different intent. It is designed to spread itself. And like most of the real "viruses" that

affect us humans, a computer virus doesn't care if the target is rich or poor, weak or strong, novice or expert. The word virus comes from the Latin for "poison", the same root that gives us virulent and viral. All denote death or grave injury.

In the first seven months of 1988, there were over 300 reports of computer virus attacks affecting over 50,000 computers. Yet less than 10% of the companies reporting these attacks were willing to publicly discuss their infections and their reactions or recovery methods. ADAPSO reported 30,000 virus infections during the last two months of 1988. Rockwell International reported in early April of this year that viruses now hit 40 to 100 of their Macintosh PCs each month.

To make matters worse, Computerworld recently published a column implying that insurance may not cover damage caused by virus attacks, particularly if no clear procedures and methods are in place to prevent or detect infection.

Fortunately, most of us have not experienced a real nasty virus attack. We've seen cutesy messages or heard about someone else having their files corrupted, but most of us are blessed by isolation -- we don't allow just anyone access to our machines, and we just don't have time to play with all the "freeware" and "shareware" that people press at us. And viruses are like our human diseases. The less contact our computer has with the outside world, the less the opportunity for infection.

Let's delve into viruses a little deeper. A virus usually leaves no obvious sign of infection of the host until after it has spread itself. You might have an infected system, but the virus often will not be active in your system until it has had a chance to replicate itself. This is particularly insidious, because it means you may harbor the viral code in several backups and not trigger the attack until, for example, you create a bootable diskette. The virus code will copy itself onto the bootable diskette, then set it's own trigger mechanism. It may only be a matter of nanoseconds, days or weeks until the trigger trips.

Viruses fall into three major categories:

- boot infectors
- system infectors
- generic application infectors

Boot infectors are viruses that replace the bootstrap code on your diskettes or hard disks. Sometimes they copy

the original boot sector(s) to other locations, but they contain enough logic to ensure that they are passed on to other disks (diskettes) that are used on the system. If a disk is accessed by the operating system and it doesn't have the virus "tag", the virus software replaces that disk's boot sector(s) with the virus software. Usually, you have to boot from an infected floppy to "catch" this kind of virus.

Back in the mid-1960's, there were several IBM mainframe "boot" Trojan Horse routines that became notorious, but they were not viruses.

System infectors usually look for a piece of the operating system, either replacing a module or attaching themselves like a piggyback rider. They may pick the command interpreter, an I/O driver or a utility command. They normally spread only when a new diskette is created with a copy of the operating system on it.

Generic application infectors are able to infect any piece of code loaded into memory or identifiable on disk as executable code. Thus, your word-processing program may look at any disk used on the system to see if it can find a .EXE file to attach itself to. Most viruses appear to fall into this category.

Some viruses are designed to work on local area networks, or hide as Trojan horses. Some have been found in vendor software, purposely placed there by employees or others intent on damaging the company's reputation. And some of us feel very safe because we don't let strangers on our personal computer, but we do take our diskettes over to the PC with the laser printer when we want the best quality printout.

So, viruses can imbed themselves in a piece of code or append themselves to the end of the code. Both of these methods make it easy to detect a change in size in the program. Some viruses also use normal system software to affect these changes, leaving a trail when the system puts a new "create" timestamp on the code. These methods leave the virus open to detection by external means. The hardest viruses to find are those that replace whole modules of code, like the boot infectors, of overlay "dead" space sometimes purposely built into vendor software (as a "patch" area, buffer or scratch pad). These are harder to find, but not impossible, given that some computers use hash totals to ensure proper loading or a module number like the Macs that increments any time the code is changed.

Most virus writers prefer the piggyback attachment method. Perhaps because it is easiest to write and work

with in the design phase. Over 99% of the code just attaches itself at the high end of a module, but a few small instructions are placed at the beginning of the module to branch into the attached code. At the end of the virus code, there is an instruction to branch back to the original code just above the replaced code. Sometimes the viral code will even restore the original low-memory instructions and branch back to the first instruction in that sequence.

In a few cases, the virus will be attached at the front-end of the code, pushing the rest of the code up in memory. This is often not effective, because the resulting module may be memory sensitive, or overlay another module at an offset from it's base that doesn't take into account the extra space required by the virus code.

Of course, writing replacement code is even harder. Not only do you have to include the virus code, but the original functionality has to be fairly well duplicated. This is not as easy at it seems, and most virus authors appear to prefer the simpler method of attachment.

What are the weak points of viruses? Yes, they have them, and that is how a whole industry of anti-virus software has grown. First, the host target must be executable code and not just data. Since most of the stored files are data, the virus must be careful not to make itself detectable by attaching itself to your Lotus worksheet file, where it would likely be highly visible, or your resume which you update monthly in Word Bender 9.16C. Because it has to process to spread, it targets executable code. On a PC, these usually manifest themselves as files with certain fixed names (COMMAND.COM), extensions (.EXE or .BIN) or attributes (stored in the directory or the first few bytes of the file).

The second weakness is that a virus always leaves a trace. It either:

- alters the size
- updates the timestamp (often both size & timestamp)
- takes up more space on the disk or diskette.

If it tries to hide as a hidden file on diskette, it still takes up space that can be detected.

Another weakness is that a virus must execute certain functions to be a virus. These are often the same functions that many application or utility programs execute, but they may occur at times or with a frequency not expected, and these can be "trapped" and monitored. Thus, it may incorporate system interrupts that give it away, or attach itself in obvious places, or contain strings of text that can be

found by searching (like the famous DEC "Cookie Monster").

There are three types of antiviral program products on the market these days, and by knowing how they work, you can put some of the ideas to work, even in a manual mode, if you have to deal with a virus infection.

- Infection prevention
- Infection detection
- Virus identifiers

Infection preventors are resident programs that monitor all system activity, waiting for telltale signs of an attempt to attach or modify code. They usually intercept or monitor the writes to disk by each program, the requests to load programs into memory, and many of the operating system services, watching for those characteristic virus patterns. Since they are always running, watching the system closely, they can impact performance and may hinder an occasional normal function that appears virus-like. They do not normally detect viruses written as replacement modules, like the boot viruses.

While prevention is probably the best policy, infection detection works by locating those common traces we talked about earlier. Has the module been changed? Has its directory entry been moved or updated? Some of these programs use a "vaccination" technique of adding a checksum or other algorythmic technique to test each piece of code for revision.

Others implement what is called a "snapshot" detection process. They compare the current state of the system with its state at the time the detector was installed. If the key parts of the system have changed, a virus is suspected. Snapshot techniques have been very successful in identifying virus infections soon after they occur. Some products are slow because they check everything in the system, but they are thorough. Others take only a few seconds because they limit themselves to the most likely areas of change (like branch address maps) to check executable programs for infection traces.

The weakness for snapshot products is that they often cannot identify infections that occurred prior to their installation.

Some infection detection programs include both the monitoring and snapshot features.

The virus detection programs products work on the source of the problem. These seek out specific virus strains or known segments of virus code and remove them from your disks. Some viruses start as labelled files or contain strings of easily detectable messages or copyright data, sequences of interrupt calls, etc, that are easy to detect, even without a special program. But the virus finders, can automate the process and remove the virus code. The problem here is that these programs work from the known world of viruses, and the designers must have seen the virus before they can detect it. New viruses may not play by the same set of rules.

What are the most likely viruses you'll run into?

1. Pakistani Brain. Developed in Lahore, Pakistan in 1986 by two brothers as an experiment. Boot sector replacement. Adds seven sectors to disk usage, infects any inserted diskette. Shows "Copyright @BRAIN" label on infected diskettes. Can cause system crash. Over 18,158 PCs infected.

2. Scores. A Mac virus originated within Electronic Data Systems in 1987. Generic application infector. Adds 7KB to program. Wakes up every 3.5 minutes. Destroys specific files, modifies icons. Over 14,207 Macs infected.

3. Israeli Virus. Hebrew University, Jerusalem, 1987. Generic application infector on IBM PCs and compatibles. Infects .COM and .EXE programs by adding 1.8KB and making them TSRs. Program files disappear on Friday the 13th. .EXE files continue to grow until too large to load. Several versions exist, some more destructive than others. Over 3,575 PCs infected.

4. nVIR. A Macintosh killer from Hamburg, German, 1987. Generic application infector, many varieties exist. Source code has been published! "Don't panic" message appears while files are being deleted. Over 11,066 infections.

5. Alameda Virus. From Merritt College, Oakland, California, Spring of '88. IBM-style PC boot infector. Does not leave much of a trail. Deletes directory entries for files or simply ignores all writes to disk. Over 3,482 PCs hit.

6. Lehigh. From Lehigh University, Bethlehem, PA, in 1987. An IBM PC & compatible "System" infector that adds 20 bytes to COMMAND.COM. Activates after four infections. Destroys all system data. Over 4,943 infections. Very vicious!

Other well-known viruses include the Columbus Day virus, which is designed after the Isreali virus, and the Shanghai virus. Europe has the PC Cyborg Trojan horse, and someone in England recently mailed out diskettes worldwide with what has been christened the AIDS virus, because it was included on a diskette containing research statistics on AIDS in humans. The diskette was clearly labeled as "may cause damage to existing computer applications."

The National Computer Security Association (NCSA) offers its members a "Virus Self Defense Kit" on a 360KB floppy. It is available to non-members for $45. NCSA is at:

Nat'l Computer Security Assoc.
4401-A Connecticut Avenue NW
Washington DC, 20015.

The kit includes a tutorial on viruses and infection detection software.

CompuServe provides a list in Library 4 of its computer consultant Forum (GO CONSULT) of a selection of antivirus software. Online Today (GO OLT 2000) also tracks viruses.

The "Safe Computing" List

In addition to the automated tools (see the addendum for a list of products known to the author), there are also some rules for "Safe Computing". There's an analogy here that prudence forbids me to mention.

1. Never boot from <u>any</u> floppy other than a vendor's original write-protected diskette. Make a copy of it, but don't boot from the copy. Using someone else's diskette, or trusting the label on a diskette you've received is an invitation to disaster.

2. One and only one bootable diskette should be kept for recovery purposes. If you don't have a hard drive, the rule still applies. One bootable diskette per PC, and keep it <u>write-protected.</u>

3. If you have a hard disk, <u>never</u> boot from floppies unless the hard disk is corrupted.

4. "Load no software before its time." Don't install 19 megs of software on your 20MB disk drive. Only keep what you need and use. Too many people have pirate copies of software "just in case" they need it, or load up on freeware or shareware to impress people with their "collection". The more stuff you load, the greater your chance of infection.

5. Treat public domain software and shareware with extreme

caution. If possible, only run it on diskette-only PCs.

6. Use subdirectories for software applications. Some viruses cannot cross the subdirectory boundaries. Keep the data is a different directory than the executable code.

7. Put internal labels on all diskettes, even if it is just the date you formatted them. Some viruses will alter the volume label, giving you a sign of the infection.

8. Watch for unusual activity. Is something running a lot slower than usual? Is there I/O to a drive when there shouldn't be? Do you have less free memory now? Do you see unusual error messages flash across your screen?

9. If transferring files on diskettes, do not include executable code or system files on the diskettes.

10. In a network environment, do not store any program files without approval of the network manager, and keep them off of common file server disks unless they are of value to everyone on the network.

11. Use the ATTRIB command in MS/DOS (or its equivalent elsewhere) to make executable programs read-only. You may have to turn off this feature for a handful of programs, but it is a simple and useful technique for a low level of security.

Recent Books

The information on this page is a synopsis of book reviews appearing 1990 issues of Computerworld:

Computer Virus Handbook, Harold Highland, $153. Includes an evaluation of 20 antivirus software packages. Wealth of material.

Computer Viruses: Dealing with Electronic Vandalism and Programmed Threats, ADAPSO, $30. 110 page, no-nonsense guide for IS professionals.

Virus! The Secret World of Computer Invaders That Breed And Destroy, Allan Lundell, $9.95. Confuses worms with viruses, but the title ought to sell lots of copies. Good history of viruses.

Computer Viruses, Worms, Data Diddlers, Killer Programs, and Other Threats to your System, John McAfee and Colin Hayes, $16.95. McAfee is a world expert on malicious code practices, and writes here about how to detect several viruses. Somewhat sensationalistic, and covers too many areas too thinly.

- -

Also recommended: The August 1989 issue of Interact, published by Interex. This issue has two good articles on viruses and several other articles on other aspects of computer security.

ADDENDUM

• Name	System	Type	Vendor
C-4	MS/DOS		Interpath Santa Clara, CA
Certus 2.0	MS/DOS	Prevent	FoundationWare Cleveland, OH
Disk Watcher VI-Spy 2.0	MS/DOS	 Detect	RG Software Systems Willow Grove, PA
Dr. Panda Utilities	MS/DOS		Panda Systems Wilmington, DE
Flu-Shot+	MS/DOS		Software Concepts New York, NY
Mace Vaccine	MS/DOS	Prevent	Fifth Generation Sys. Baton Rouge, LA
SoftSafe	MS/DOS		Software Directions Randolph, NJ
Vaccine	MS/DOS	Detect	WorldWide Data Corp. New York, NY
Virusafe	MS/DOS		Comnetco Inc. Bernardsville, NJ
Virus Guard	MS/DOS	Prevent	IP Technologies Coste Mesa, CA
Vir-X	MS/DOS		MicroCraft Richmond, IN

CA-Unipack/SCA	IBM MVS	Detect	Computer Associates Garden City, NY

ADDENDUM

• Name	System	Type	Vendor
Antiplan 1.1	Mac		unknown
Disinfectant	Mac	Detect	unknown
Ferret 1.1	Mac	Detect	unknown
Interferon 3.1	Mac		unknown
SAM 2.0	Mac	Detect	Symantec Cupertino, CA
Virex 2.5	Mac	Detect	HJC Software
Virus Rx	Mac	Detect	Apple any dealer

PERFORMANCE MANAGEMENT - THE PEOPLE KIND

Lynn A. Novo, CPC
Network Systems Company
10 Diana Court
Cheshire, CT 06410
203-250-1919

INTRODUCTION

Evaluation and appraisal of your data processing staff performance is as important as monitoring your system performance.

Performance management is the tool that will help managers produce HP professionals who are willing to make a meaningful contribution to the department's success.

We define performance management not as the required documentation for annual salary reviews, but an ongoing process of communication between manager and staff about on-the-job progress.

"But this involves time and I don't have time!" You don't need much time. A positive attitude, an enthusiastic approach and some simple techniques will yield the results in increased performance you hoped for.

This paper will discuss why performance management is so important to the success of the HP department. We will also outline specific "how-to's" that work for other managers.

TARGET SHOOTING

The creative nature of any data processing job makes it difficult to set standard parameters of measurement. So how can you evaluate DP personnel? Is it really necessary?

An operator at a local HP shop said "If I'm doing a good job, I won't hear from the manager, if I'm not, I will!" This gunshot approach is just not effective for most DP personnel.

It is extremely difficult for your staff to increase productivity, or be more motivated on the job, if the quality of their current performance is unknown.

Consider this:

Imagine a person who is passionately devoted to rifle shooting. He lives only for the sport of target-shooting on the rifle range.

Imagine this man receives a cash windfall that allows him to devote the rest of his days exclusively to his favorite sport - but under the following conditions:

He may shoot as much as he wants - but the score will never be communicated to him. Thus he will never know if he hits a bull's-eye or if he is even close to it. He is denied the information he needs to correct his aim and improve his score.

Can you imagine anything more senseless or demoralizing for a marksman? He will, of course, be aware that he is not always hitting the bullseye; he doesn't expect to - but with no feedback at all on his performance he will eventually lose interest.

If the acknowledgment of results in an avocation is so important, how much more important it is in the workplace, where we spend most of our time.

What are HP managers actually doing about performance management?

We asked several of our clients how they managed performance of their DP staff.

"Our performance review is like honoring birthdays. We have an annual celebration, to look back at our achievements and realize, we didn't do so bad."
MIS manager - local university.

"The best employee in the department will get the best pay increase out of the amount allocated to the department. We determine the best employee by an annual corporate review."
DP department supervisor - aerospace and defence.

However not all HP staff think their managers are doing a good job.

"The only way to get an increase in pay is to move into a new position. We don't get appraisals."
Programmer - computer & telecommunications firm

"We are supposed to get merit reviews. I think it's a joke! Everyone knows no one has ever had one. As for salary, we all get a flat 4% increase."
Systems analyst - division of food, drink and tobacco processors.

"Our performance appraisal consists of a one year review. However I don't know why they bother. Everyone gets the same salary increase, it is not based on merit. The company loses out because it doesn't foster any competition between us, so why perform?"
Senior programmer - consumer products manufacturer

From the mixed answers we heard we can conclude that some companies are trying to encourage performance management. However they seem to be missing the mark with DP professionals. What may work on marketing, production or administrative departments, will not necessarily work for the MIS department.

It is obvious DP personnel are special; a different breed, requiring a new approach to review techniques. A motivated, productive, satisfied workforce is possible if attention is drawn to the unique needs of information systems professionals.

ZAWACKI & COUGER: IS TRAITS

Robert Zawacki and Daniel Couger are two researchers who have compiled a great deal of information on what makes IS people "different." They believe that matching the right job with each person on your staff is essential for their success.

We will take a little time to explain the basics of their study, as it will help clarify how we can improve performance management of your staff.

Matching jobs to IS professionals requires knowing the degree to which a job contains certain core job dimensions that motivate as individual intrinsically. This "MPS" (Motivating Potential Score) is coupled with the measurement of your employee's need to grow and develop the "GNS" (or Growth Need Strength.)

The only problem we encounter is that the technical environment and different tasks involved makes a standard measurement of DP personnel complex. It is best to evaluate staff in job categories. Do you measure your programmers by the number of lines coded? Or how well the documentation is done? How do you measure the creativity involved in writing a program, or finding the bug?

It is difficult to use quantitative measurements all the time, because everyone excels at something but not everything. The most equitable and viable way is to define and differentiate between performance standards, goals and objectives.

Zawacki defines these variables as:

Performance standards: specific statements expected of the individual within a relatively short time period i.e. daily, weekly or monthly.
Goals : intermediate statements of results people desire to achieve in time frame of one month to year
Objectives : results the company wishes to accomplish during time frame of one to five years.

Taking these variables and considering the match-ups of jobs with personality traits already explained, we can offer some guidelines:

* "Key job requirements" can be defined for each DP member. What the individual wants in the job, what the manager wants to see done and what is viable to do. These should be discussed regularly and be reestablished on mutual terms if necessary. "Key job requirements" are useful as a base for the annual salary reviews as well.

* Evaluate the nature and the scope of the job the individual is hired to do. Individuals with the need for a challenge require a job with diverse skill variety and one that is perceived as meaningful.

* DP managers must strike a happy medium between rigorous, complex management technique and neglecting their managerial role. A simple but well structured style can give you the results you desire.

* Translate feedback into specific goals. Make the goals challenging but reachable, in a relatively short period of time.

WHY MAKE TIME TO ASSESS DP PROFESSIONALS?

Performance management can often seem like a waste of time and effort, with no visible results. "My programmers don't need it." "I am too busy." "The company annual review is enough." So why should I bother to make extra time and do it?

As Zawacki and Couger demonstrate, DP professionals want it and need it!

What can management of staff performance achieve? It is part of your managerial duties. A well managed staff with clear two-way Bcommunication with the manager is the basis for a positive, creative work environment. Mutual goal setting leads to a stable, motivated, productive workforce. But what else can it do for the individual, the department and the company?

* Open and honest personal feedback can help identify ways to assist the individual's own development. This can lead to improved performance on the job and create opportunities from changes occurring in the job. It could produce a team who are proud of their achievements, demand more from themselves and require less management.

* Progress is reviewed and future development of the individual can be considered, both in overall and potential performance.

* Management of performance can:

- improve communication between the individual and managers
- create an opportunity for greater participation in future planning of work-loads and target setting
- provide a solid basis for succession and manpower planning
- organize training requirements
- formulate fair and logical salary policy.

Managing staff performance, if approached correctly, helps keep the manager in touch with staff, on a more positive note.

A motivated workforce is one ingredient of increased productivity. If the source of motivation lies in successful results, it is important to acknowledge these results. How do you do this? By effective performance management.

Believe it or not, those who master the concept of effective performance management can benefit from reduced stress. Collaborative goals are set, you have confidence the individual will achieve these goals, productivity is increased and stress level is reduced. More time will be available for other managerial tasks and a more positive working environment has been created.

THE "HOW TO'S" OF PERFORMANCE MANAGEMENT

We now know the benefits of performance management, but where do you start?

READY - audit your resources and prepare information
AIM - identify needs and plan future action
FIRE - take the action to hit the performance target
SCORE - review performance results and provide feedback
ADJUST - counsel on performance, changes and improvements

The first step is to assess how you are doing right now. Is the department meeting its requirements? If not, why? If you can establish or identify a problem area early on, it will be easier to rectify it.

You must develop a strategy that will help assess performance levels. Make a plan with visible goals and deadlines; have something to aim for.

Start to keep a confidential file on each staff member. Successes and problems can be documented for easy reference. The more you get to know about someone the easier it will be to manage them and reach the productivity levels they can achieve.

HELP FROM THE TEAM

Create project teams often. Divide systems analysts, programmers and others involved into small teams. Assign a project leader to each team. Meet on a monthly basis, depending on development

time, to discuss the ongoing work and any problems encountered.

Ask a member of the group to submit reports or documentation, each time you meet. Let them identify any problems and give their impression of how the project is progressing. You can cross reference your results and see how they correlate. Are your thoughts and ideas consistent with the team's?

This approach can separate key players in the department from those just going along for the ride. Sometimes it is difficult for managers to assess this on their own. Openly praise staff in front of other team members to boost morale. But if you spot a problem member, take the individual aside and speak confidentially.

Tracking performance levels of your staff will help to separate the skilled from the unskilled. When you are looking for the next project leader, you will be able to identify your best candidate, quickly and efficiently.

PRACTICAL TIPS FOR MIS DEPARTMENT APPRAISALS

In order to provide a positive atmosphere for performance feedback, managers only need a few guidelines.

1. Be friendly and approachable.
2. Establish performance standards for each staff member.
3. Document the good and bad for each staff member on a regular basis.
4. Recognize good performance and critique areas that need improvement.
5. Maintain a supportive work environment.

Factors influencing the management techniques used and how the staff performs will be affected by:

* corporate culture
* size of the department
* scope of the job being evaluated

Performance management needs to be customized to each individual member of your team. Those individuals needing more attention and direction will be managed very differently than those who thrive on minimal contact.

One manager we know has his staff summarize the week's activities, including things that went well and things that caused problems. The highlights are reviewed by the manager in a 15 minute (or less) weekly meeting. The manager adds his own observations from the week. This includes 'pats on the back' as well as specific recommendations to correct or improve problem areas.

"I like to get my staff involved and help them see the benefit in what they are doing. I do my best to get to know each team member and their interests. My policy is to praise and appreciate people with honest sincerity."

In such an environment, conditions for managing performance could not be better. For once the subordinate is the center of attention. Open discussion can reveal how well the individual feels suited for the job, what they want to do next in their career and how far within the company the individual wants to go.

DO I NEED TO DOCUMENT?

Documented information is an essential record of performance history. Without this, effective and enforceable decisions about pay, discipline and promotion, cannot be made. Such a record helps to evaluate and compare individuals both at present and in the future.

The key is to make this simple. Formal documentation is not necessary all the time; keep notes on each individual and consolidate them every once in a while.

Every DP department will have different team members and needs. The size of the DP department will obviously influence the managerial style. The larger the department, the more documentation required to differentiate one staff member's activity from another.

MBWA

Managing performance is an ongoing process and can be a key requirement in effective annual salary reviews. Often the individual sees the formal review only as an evaluation for a pay raise or promotion opportunity. Unfortunately this is the case in many companies, but performance management as we define it is a motivating, investigative, management tool.

Regular meetings will help track how the individual is progressing and identify any weak spots before they get out of hand. Frequent meetings are more advantageous than a single annual review, however informal.

Performance management sounds like a lot of hard work, but it can be as easy as "Managing By Walking Around" (MBWA): Tom Peters style - "find out what's going on in the field. Catch people doing things right."

Here are some good reasons for MBWA excerpted from Strategic Management by Thompson Strickland.

"Once a division or department has developed a plan of its own -a set of working objectives - it's important for managers and supervisors to keep it in operating condition. This is where observation, measurement, feedback and guidance comes in. It's our "management by wandering around." That's how you find out whether you're on track and heading at the right speed and in the
right direction. If you don't constantly monitor how people are operating, not only will they tend to wander off track but also they will begin to believe you weren't serious about the plan in the first place. It has the extra benefit of getting off your chair and moving about your area. By wandering around, I literally mean moving around and talking to people. It's all done on a very informal and spontaneous basis, but it's important in the course of time to cover your whole territory. You start out by being accessible and approachable, but the main thing is to realize you're there to listen. The second is that it is vital to keep people informed about what's going on in the company, especially those things that are important to them. The third reason for doing this is because it is just plain fun."

During these meetings you should find yourself making a few promises. Your senior programmer wants to move into management, but needs to learn some business skills first. Investigate company policy on tuition allowance, time off for education, etc. Get him enrolled and involved in the course he wants. Show your support for his enthusiasm. Is further training required or requested by one of your operators? If so, get them the training. False promises quickly lead to a demotivated, unproductive workforce.

Performance management can uncover problems you were not aware existed. Is John interfacing more with the users, as you suggested? Don't ask him out right. Observe him, ask the users. If positive results are not visible it is time for further performance actions.

Each individual staff member will need different attention and feedback. Assess the value of the meetings and make them appropriate to your needs, the staff member and department productivity.

CONCLUSION

Like the target shooting devotee, HP professionals need to know the target they are aiming for and they need to know how close they are to hitting a bullseye.

Set performance standards, goals and objectives for your deBpartment and each individual. Then catch your staff members doing something right and make a note of it. Offer critiques of problem areas with constructive suggestions for improvement. Set a deadline to meet again to review performance and set new objectives. Before long, you'll see that you and your staff have established a habit of on going communication about on the job performance!

For Further Reading:

Blanchard, K and Johnson, S. The One Minute Manager. William Morrow, 1982.

Brennan, J. Performance Management Workbook. Englewood Cliffs: Prentice Hall, 1989.

Yate, M. Keeping The Best. Holbrook: Bob Adams, Inc., 1991.

Zawacki, Robert. "Performance Standards, Goals and Objectives for DP Personnel." Journal of Systems Management, January, 1984

Zawacki, Robert. "How To Keep Eagles" ComputerWorld, July 27, 1987, pp. 51-57

Zawacki, Robert. "The Real Key To MIS Motivation Is In The Matchups" Business Month, March 1989, pp. 77-78.

Zawacki, Robert. "Microwave Management" Manager's Journal ComputerWorld, March 18, 1991.

TITLE: Linking Corporate Strategy And Information Systems

AUTHOR: Dick Onel

DCE Information Management Consultancy

Groenezoom 1
Badmoevesorp, 1171 JA

HOLLAND

+2968 97751 Fax +2968 97076

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPERNO. 5152

5153: A Practical Approach To Disaster Recovery Planning

John Painter
Computer Solutions, Inc.
397 Park Avenue
Orange, NJ 07050
201-672-6000

Introduction:
Recovery planning is no easy task. The process is very often time consuming, it's difficult to organize, it requires interaction with many people in different departments, it may raise more questions than it answers, and the reward is only a document that will sit somewhere on a shelf, probably never to be used in a real disaster. What fun.

Actually, recovery planning can be made a bit more bearable if a practical approach is taken to the task. There is a simple set of steps that can be used to organize and direct the planning process. There are principles by which to determine acceptable risks. There are lessons to be learned from the experiences of those who have gone before you. There are tools to make the process move along at a quicker rate. These steps, these principles, these lessons, and these tools can be integrated into a cohesive whole to form what I call a "practical approach" to recovery planning. This approach will yield a recovery plan for your organization that is comprehensive, viable, and relatively painless to prepare. Before we examine this approach in detail, lets quickly cover some of the basics about disaster recovery.

The Recovery Issue
There are a variety of reasons that disaster recovery has become such an important issue among those charged with maintaining organizational information services. Today the primary functions of most businesses are computer-run and management has become more aware of this dependency. Management has also become aware of their accountability for these information systems. The Foreign Corrupt Practices Act of 1977 requires the maintenance of complete and accurate records of all business transactions. For this reason, among others, there is internal and external audit pressure for recovery planning. Many auditors require a written plan and a hotsite or coldsite. Some go as far as to inspect the hotsite facility themselves. Some industry regulators-- such as in the Banking industry-- also require recovery planning. Finally, recent disasters like Hurricane Hugo, the California earthquake of 1989, and the Wall Street power failure of 1990 have focused attention on the risks of being caught unprepared.

The Wall Street disaster is an excellent example of being caught unprepared. Some organizations assumed that Manhattan would be relatively secure from large scale disasters. Weather is rarely a problem and things like earthquakes and tornados are unknown. However, on August 13, 1990 a fire struck a Con Ed power substation in the South Street Seaport area. The fire knocked out power for 36 square blocks and affected 320 data centers. To date this is the largest single disaster in terms of the number of data centers brought down. The Federal Reserve Bank of New York-- the clearinghouse for checks around the nation-- recovered at their IBM hotsite. The American Stock Exchange, the New York Mercantile Exchange and the New York Commodities Exchange were all closed for some period of time. Most companies had some sort of recovery plan in place-- but many felt the inadequacy of these plans. Some organizations that had counted on their own backup power found that their expensive diesel generators had not been maintained well enough or that their generator was being asked to handle too heavy a load. One organization found that, although their computer center had backup power, their building did not. Therefore, the fire department could not allow occupancy of the building because the fire alarms would not work. The data center was operable, but no one could get in the building to use it! So much for that recovery plan.

Recovery Choices

There are four basic means by which an organization can recover their data center operations. Each has advantages and disadvantages, and each is appropriate for a different level of risk avoidance. The first and most reliable means is a hotsite agreement. A hotsite is a fully equipped computer room with the computer(s), the peripherals and the communications to act as a backup to your own site. In an actual disaster, a subscriber to a hotsite would load their last backup tapes onto the hotsite system and have their users access the hotsite system from the actual hotsite or remotely by using dial-up systems, networks, or some other means. Previous testing would ensure compatibility and adequate capacity of the hotsite system. This means of recovery is the most reliable because the backup system is always available and it has been configured to suit the subscriber's needs. Presumably at least one test of the hotsite system has been successful, thereby ensuring that no surprises will occur during an actual disaster. The disadvantage of a hotsite is the cost. Hotsite vendors ask that you pay a fee, usually on a monthly basis, to reserve your backup system and communications. Other fees are associated with the act of invoking a disaster. Since the hotsite vendor is required to keep the hardware idle, the expense is born entirely by the subscribers. Still, the cost of a hotsite can pale in comparison to the losses that may occur due to extended downtime.

A second means of recovery is a coldsite agreement. A coldsite is empty computer room space provided with raised flooring, power, and air conditioning. Coldsite agreements are inexpensive in comparison to hotsite agreements, but they don't

offer the same level of risk avoidance. In an actual disaster, the subscriber would be responsible for locating and configuring a backup system to use in the coldsite space. If your own system is not damaged, recovery may only mean relocating your system to the coldsite space. If your system is lost, however, than it may take several weeks and lots of money to set up a backup system in the coldsite. Communications may also take some time to be established. Coldsite agreements are an inexpensive way to reduce a portion of your risk exposure, but they may require that your organization be without a functioning system for several weeks or more.

The third means of recovery is a portable system. Portable systems may come in the form of a freight-ready bundle that is to be shipped to the designation of choice, or in the form of a trailer with a mini hotsite on board. The advantage of this means is that communications can be simply re-routed at the subscriber's site from the existing system to the portable system. However, this can become problematical if the subscriber site cannot be occupied due to the nature of the disaster or if the site is located in a building or area not easily accessible by a tractor trailer. Portable systems rival or exceed the cost of comparable hotsite agreements. They are quite useful under certain physical site conditions and under certain disaster scenarios, yet unworkable in other situations.

The fourth means of recovery is the reciprocal agreement. This is an agreement between two or more sites, usually in the same organization, that would allow one site to share the use of a system in the event of a disaster. This sort of agreement was popular when recovery first became an issue but it is rarely used anymore. Today, most organizations are using their system(s) at or near capacity and there is very little possibility for a reciprocal agreement to work. In fact, most reciprocal agreements are non-binding and there is no guarantee of any access to a system at all. This is a clear case of getting what you pay for. Another version of this means of recovery is an agreement with your hardware vendor to provide the "next machine off the assembly line" in the event of a disaster. Again, you are at the mercy of the vendor and you have little guarantee of a stable recovery.

Categories of Disaster

Now that we've taken a look at the means of recovery, let's examine some of the forms of disaster. Experts group the causes of disasters into four broad categories. First, there are the scenarios we typically associate with disasters: fires, floods, wind damage, earthquakes, hurricanes, etc. We'll call these "Catastrophic" disasters. The second category involves accidents like plane or vehicle crashes, explosions, and hazardous material spills. We'll call these "Accident-Related" disasters. The third category includes equipment failure in the computer center or end-user location, failure of communications lines or networks, and breakdown of electrical service or plumbing. We'll call these "Equipment-Related" disasters. Finally, there are

disasters caused by human action: sabotage, employee or non-employee strikes, terrorism, boycotts, protests, and acts of government. We'll call these "Human-Action" disasters. While many organizations plan effectively for the first two categories, the last two categories are very often ignored. This is ironic because experience shows that Equipment-Related and Human Action disasters are usually more likely to occur.

Recovery planning should be as much about unexpected, small-scale interruptions in operations as about major natural catastrophes. Truly effective planning is focused more on the everyday risks facing your organization than on the less-likely risks of large-scale natural catastrophes. Certain geographical areas are obviously more susceptible to natural disasters than others, but it is crucial that ample consideration be given to the "mini-disasters" that are the most likely to affect operations. For a data center, the definition of a "disaster" is much wider than merely earthquakes and fires-- consider some common "mini-disasters":

Accidents: Manufacturing and some corporate centers are often located near major highways and railways. These centers can be affected by train or car wrecks and by hazardous material spills.

Building Issues: Inspect the air conditioning and heating systems--redundancy is always desirable in the event of a failure. Be wary of leaking roofs. Determine if your building contains asbestos that may require removal or containment.

Data Communications: Keep spare modems and multiplexors. Maintain a diagram of all communications, including any networks. Consider running some data lines to different COs or subscribing to a microwave based communications recovery plan.

Plumbing: Check for plumbing pipes running overhead or through the computer room walls. See that old sprinkler systems are properly drained.

Power Problems: Besides the obvious risk of brownouts and blackouts, there can be power problems within the site or within the system. Proper maintenance, line conditioners and backup power sources can reduce these risks.

Sabotage: Disgruntled employees can go to extremes. Computer room and building access should be restricted and monitored. Termination procedures should be strict and thorough. Passwords should be changed often.

Security: Site security should include procedures to handle strikes, protests, building evacuation, bomb threats, and uninvited guests.

Software: Be aware that upgrading current applications or operating systems to new, untried versions can bring a system down. An alternate system may be needed until the upgrade is running properly.

It's easy to get caught up in the process of recovery planning and lose track of the original goal. In order for an organization to properly reduce the likelihood of an unrecoverable disaster, it is necessary to be realistic in the analysis of risks. Too many organizations end up with a huge, multi-tiered plan to recover from Catastrophic disasters and get caught by the little things they never considered. It is critical to examine the most likely risks facing your organization and build your recovery plan from there. It makes little sense to clutter your recovery plan with disaster scenarios that are highly unlikely to occur. There must be a point at which the planner says "enough."

Defining Recovery

We've examined now the means of recovery and taken a brief look at some likely forms of disaster. Let's take a minute to define what we mean by "recovery". There is an advertising slogan that has been brandished about the disaster recovery market for years-- "BUSINESS AS USUAL!" This slogan has seemingly infiltrated the minds of many recovery planners and caused an unusual amount of unnecessary misery. This is because the Business As Usual slogan is a myth. Those who try to plan an organizational recovery around the notion that the recovery period will be business as usual will come to realize two things. One, the attempt is going to cost a lot of money. Two, it can't really be done anyway. Even if a recovery goes smoothly according to plan, changes are necessary to make a recovery work. Communications will be re-routed. Operators will be working at a different location. Users may be inconvenienced or asked to change their normal routine. Some applications or databases will not be available or access will be sharply curtailed. The fact is, even a duplicate hotsite system with a complex data mirroring scheme would require some changes to be made in a disaster-- and the vast majority of organizations have no where near the budget to build such a system anyway.

Recovery planning becomes instantly easier when the planner recognizes that the goal of a recovery is business continuity, not a regular day at work. Recovery planning should aim to control any crisis from spiraling out of control and provide a stable path to full recovery. This means that applications are prioritized and only the critical applications are recovered. In turn, the backup capabilities are often less then that in the normal day to day operations. If an organization normally runs a

full blown HP 3000 Series 960, but has only one critical application in a disaster scenario-- odds are that organization can get by with a modestly configured Series 950. Planning becomes more practical, realistic, and workable when the business as usual hype is left behind. This principle, in conjunction with a realistic analysis of the most likely risks, will make the planning process more manageable and the resulting plan more practical.

Four Stages of Recovery Planning

So far we've discussed why disaster recovery has become an important issue, we've examined some of the categories of disasters and the likelihood of each, and we've studied some principles by which to make recovery planning a more realistic process. Now we're in a position to create a structure for the planning process. Recovery planning can be broken down into four fundamental steps.

The first step in recovery planning should always be to determine just what applications are critical to maintain. The principal goal of a disaster recovery plan is to provide the ability to quickly restore service for critical applications when there is a serious failure or disruption of regular operations. Prioritizing applications is a first step toward establishing what your minimum backup requirements will be. An application may be considered "critical" if its loss would significantly and adversely affect current or future revenues, business records, or immediate research and development plans. The priority of critical applications should be determined according to a relative scale of importance to the continuity of the organization.

Keep in mind that a practical recovery plan must be sufficiently detailed so that major decisions will have been made prior to, and not immediately after, a serious situation occurs, but not so detailed as to be ineffective in an emergency. Thus, prioritizing your applications helps ensure that, first, no critical needs will be missed, and second, that your plan is not so detailed as to be useless. Prioritizing also will help you to determine what your minimum hardware configuration would need to be, should you decide to subscribe to a hotsite. As we mentioned earlier, your backup system will almost always be less capable than your actual system. You are seeking the ability to recover critical applications, not to duplicate your current system.

The second step in effective recovery planning is to define how long your critical applications can be down. If an organization can live with a week or two weeks of down time, then it can ignore many potential risks and threats. A 24 hour power outage, for instance, would only be an inconvenience. On the other hand, if an organization cannot be without it's data center for more than 12 hours, then that same power outage becomes a disaster. The amount of acceptable downtime is perhaps the most crucial aspect of recovery planning.

Knowing the amount of acceptable downtime will also help you decide on your means of recovery. For instance, if your acceptable downtime is two or three weeks-- then a coldsite agreement would be in order. Two or three weeks might well be enough time to locate and install a system and appropriate communications in your coldsite space. If, on the other hand, your acceptable downtime is a few days or a number of hours, then a hotsite agreement would be necessary to ensure a stable recovery.

The third major step in recovery planning is to determine the potential risks and threats that could affect your data center past the point of acceptable downtime. In other words, you must compile a list of the disasters likely to face your organization and weed out the scenarios that would bring down your system for a longer period of time than your organization can endure. The possibility of large-scale natural disasters should be investigated if your geographic area is susceptible, but the smaller-scale risks that may be more likely to cause a problem should not be overlooked. As noted earlier, it is the "little things" that will bring down computer operations more often than not.

Use the four categories of disasters to help analyze the risks and threats facing your organization. Be thorough, but don't dwell on fantastic scenarios that are extremely unlikely to occur. The more realistic your analysis is, the more practical, workable and affordable your plan will be.

The fourth major step in recovery planning is to find methods to control the risks and threats that could bring your system down for a longer period than is acceptable. Most organizations will find that a hotsite or a coldsite is necessary. A recovery plan that defines command and control procedures in an emergency and that defines the general steps to be taken in an emergency is a must. There are also any number of other simple and affordable steps that can be taken to reduce risk exposure.

Some examples make this clear. The risk of sabotage can be reduced simply by tightening existing security procedures. Make sure guards stay at their posts. Restrict computer room access. Change passwords often and maintain strict termination procedures. The risk of fire damage can be reduced by inexpensive alarms and simply by keeping paper and other supplies away from the data center. The risk of plumbing breaks can be minimized by keeping a cheap plastic tarp in the computer room which can be tossed over vulnerable hardware in the event of a water leak. One computer operator who has been through a serious power outage tells the story of setting his tie on fire while using a Bic lighter to navigate his way through a dark computer room. Simple items like penlights and emergency toolkits

can make the difference between an easy or difficult initial recovery period.

Available Tools

The four fundamental steps in recovery planning are a useful way to structure the planning process. Designing your own recovery plan has the advantages of custom fit and high employee involvement. The disadvantages are the length of time required, the high learning curve, and the possibility of hit and miss organization. After all, it is up to the planner to take the initiative to learn the issues and organize the plan. However, other tools are available to help you put together an effective plan. There are many commercially available recovery planning packages from which to choose. Most are PC based and many are extremely easy to use.

There are two main types of PC based planning packages; one type has a database structure, the other type has a word-processing structure. Some hybrid packages are available too. The word-processing based packages are generally cheaper-- ranging between one and five thousand dollars. The database structured packages range between three and forty thousand dollars. Word-processing based packages are generally easier to use and are appropriate for almost all small to medium sized data centers. The database type has a more complex structure, it can handle larger volumes of information, and it is appropriate for large data centers with multiple systems and multiple sister sites. The advantages of using a PC based plan are less work, a more coherent end result, ready-to-use forms and structures, and having the help of a road map to follow. The disadvantages are that the planner must first learn the package and remember the package in order to update the plan. Overall, though, most harried MIS recovery planners would benefit from going the route of a commercial package. These packages will not do the planning or make the tough decisions for you, but they will organize the process and provide you with the tools to design a comprehensive and effective plan.

The Elements of a Good Plan

Whether or not you use a commercially available plan or design your own, there are a number of items that should appear in every recovery plan. Let's run down a short list of must-haves.

Every recovery plan should establish a recovery team that would be activated in the event of a disaster. This team becomes the command and control mechanism to implement the recovery plan and make the quick decisions that need to be made. For smaller organizations, team members might only include a handful of MIS people. Larger organizations will include members from other departments such as insurance and corporate management. Some organizations are large enough to require multiple teams, with each team assigned a specific task. The recovery team members should be knowledgeable about your organizational requirements, be

available on short notice, and have the authority to pursue the means necessary to ensure full recovery. The team should meet on a regular basis and each member must be made aware of updates made to the plan. The recovery team is essential for smooth recovery.

There are a number of lists that are helpful to have in an emergency situation. A list of suppliers and vendors should be included in your plan, complete with addresses, phone numbers and a contact name. Include your hardware vendors, your forms and supplies vendors, your software vendors, your maintenance vendors, your offsite storage vendor, and any other vendor or supplier that might affect computer operations. A list of alternate vendors is also helpful. In an emergency, it may not be possible to find the employee who knows where your organization buys a certain item. The vendor list ensures that hardware and supplies will be available.

A list of all hardware with serial and model numbers should be included for insurance purposes, and a detailed floorplan of your data center should be filed right alongside. A list of applications-- prioritized according to relative importance-- should be included for reasons discussed earlier. It's a good idea to make a copy of the front page of each application manual too. This page will have the version number of the software and will offer some proof of ownership if the software or the manuals must be replaced.

Include copies of all your hardware maintenance agreements, support agreements, and software license agreements. These could be unavailable depending on the type of disaster. Include a library of any custom forms or paper your organization may use so that they can be replaced easily. Keep a copy of operating procedures and special logbooks too-- these will be valuable for use in hotsite or coldsite operations. Finally, include a copy of your hotsite, coldsite or reciprocal agreement and the information necessary to use the agreement if needed. Directions to the hotsite and notification procedures are two examples.

Overall Strategy

There are a few things to keep in mind throughout the planning process. First, keep the analysis of risks and threats on a realistic level. Don't fall for the Business As Usual myth. Second, walk the line between complexity and utility. If a plan is too complex or too simple it will be of little use in a real emergency-- try to find a middle ground. Third, take the plan seriously. Assign competent personnel to construct the plan. Make upper management aware of recovery issues. Schedule regular updates to the plan. Keep multiple copies of the plan in scattered locations. Finally, test the plan. Do at least one hotsite test per year, schedule mock disasters at your site, and make certain the recovery team members understand their role in the recovery process.

Review

We stated in the beginning that recovery planning can be made more bearable if a practical approach is taken to the task. The practical approach I'm speaking about starts with a clear understanding of what disaster recovery is all about and what the choices for recovery are. Disaster recovery is about maintaining business continuity, its about keeping the computer applications that are critical to your organization up and running. The principle of business continuity rather than business as usual must guide the practical recovery planner. The practical approach is also dependent on an informed view of just what constitutes a disaster. The voice of experience tells us that disasters are not always the natural calamities we typically call to mind. Many other "mini-disasters" are as likely or more likely to bring computer operations to a halt.

The four stages of recovery planning-- defining critical applications, determining maximum downtime, analyzing risks and threats, and finding methods to control risks and threats-- provide a structure to the practical approach. These four stages help to ensure that minimum needs will be met, but that unnecessary or unrelated affairs will be left out of the equation. We've seen that a practical approach will take advantage of available tools like PC planning packages and will learn from the experiences of others. Finally, this approach takes note that every recovery plan shares a number of required features like recovery team members and certain essential information. The practical approach to recovery planning, if taken to its logical conclusion, will result in a comprehensive yet workable and affordable recovery plan that might one day be vital to your organization.

TITLE: Tracking Software Quality Using Standard Questionnaires

AUTHOR: Rick Paquette
Weyerhauser
WWC - 1W6
Tacoma, WA 98477
(206) 924-3109

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 5154

PAPER NUMBER : 5155

LET GO OF THAT BANANA!!

NANCY PEACOCK
BIO-RAD, INC
1000 ALFRED NOBEL DR.
HERCULES, CA
415-724-7000

More and more, stress has become a major career buzzword. Articles on stress appear in the trade journals, stress workshops are popping up at all of the conventions, and companies are bringing in experts to train employees on coping techniques and methods.

So ... what's all the noise about? What is stress and why should you be concerned? What can you do about stress, anyway? These are the main topics addressed in this paper.

WHAT IS STRESS?

The stress reaction is actually a survival mechanism. It is designed to engage all of your faculties in preparation for either fighting or running away. When we perceive a threatening situation, more than fourteen hundred physiochemical reactions occur in the brain and body. Most of these reactions are delivered through the involuntary nervous system and the neuroendocrine system. Some of the more obvious changes are:

. Blood rushes to the muscles to prepare for action
. Breathing increases to supply oxygen to the blood
. The heart rate accelerates to pump more blood, increasing blood pressure.
. Sugar and fats are dumped into the bloodstream to provide energy
. Blood clotting mechanisms are activated
. Non-essential processes, such as digestion, shut down.
. Hormones, including insulin, hydrocortizone, and adrenalin are secreted into the blood
. Gastric acid and blood cholesterol levels rise
. The body's immune response is inhibited.

These reactions come in stages, and become more and more harmful over time if the stress continues:

First, the pituitary and adrenal glands are stimulated to release hormones that mobilize the body's defenses. The autonomic (or unconscious) nervous system gets the body ready for extreme action and for restoration. Once set into motion, this system functions outside of conscious control.

Next, if the stress continues, hormonal secretions are activated to counteract the effect of the first stage, providing the negative feedback necessary to keep things in control. Unnecessary body functions are shut down. The body's defense against other stressors is significantly weakened.

After extreme continued stress, the body may become so vulnerable from all of these changes in normal metabolic activity that all resistance breaks down and exhaustion ensues. The body cannot even respond to simple infections.

The stress reaction was appropriate when the major threat was something like a charging rhino. Even now, if a problem can be solved through "flight-or-fight", the reaction proceeds to its intended conclusion. The chemicals and hormones that were released are used up; your body recuperates and returns to a state of equilibrium. The solution itself dissipates the stress symptoms. This is called short-term, or acute stress.

Our current technological environment, however, rarely affords the chance to deal so directly with perceived threats. Most daily threats are ambiguous. There is no way to react and recover. The result of stress without action is called long-term, or chronic stress.

Chronic stress arises when the mind decides that a physical reaction to the perceived threat would be inappropriate. The chemicals that were produced remain in your bloodstream and your body remains on the defensive, charged for action. These chemicals, necessary in the short term, are not intended to stay in your body. If allowed to remain there, they cause permanent damage:

> Chronic high blood pressure
> Protein imbalance
> Inhibited immune response

... all of these age your body and increase your vulnerability to disease.

THE IMPACT OF CHRONIC STRESS

Society pays a heavy toll for chronic stress, and so do you.

Stress-related conditions are bogging down our insurance companies with claims, reducing the effectiveness of our work force, and robbing us of our own enjoyment of life. Consider the following statistics:

>An estimated 3 percent of the American work force is absent every workday of the year.

>Lost productivity due to absenteeism is estimated to be as high as $150 billion.

>Stress related disorders account for 54 percent of the absences

>Even when on the job, only 2 out of 10 employees work to their full potential, according to the National Commission on Productivity.

>Most standard textbooks attribute 50 to 80 percent of all diseases to psychosomatic or stress-related sources.

On a more personal level, how many of your family, friends, and acquaintances suffer from hypertension, chronic pain, migraines, substance-abuse, or other stress-related diseases?

Stress has been shown in basic and clinical research to be a major factor in a wide range of conditions, including:

> hypertension
> cardiovascular disease
> gastrointestinal disorders
> tension and vascular headaches
> lower back pain
> decreased immunological functioning

Stress has also been shown to increase other health-related behaviors, such as smoking, alcohol consumption, compulsive eating, and even mental disorders.

Stress can bring on exhaustion, disease, and even premature death. The cost of stress to you personally is paid out not only in dollars and cents, but also in decreased life span and reduced enjoyment.

SOURCES OF STRESS

LIFE CHANGES:

Life changes are among the most demanding situations we encounter. Changes in our family composition, employment, residence, or finances test our ability to adapt. We are often confronted with the need to accept situations over which we have little control.

INTERPERSONAL RELATIONSHIPS:

Another area in our lives that provides potential stress is our interpersonal relationships. Our thoughts and feelings about ourselves and others, and the extent to which we can express them, contribute to the degree of comfort or tension we feel in most situations.

TIME ISSUES

Time and how we relate to it ultimately colors our thoughts, feelings and actions. Putting things off until the last minute, always being a little late and asking for more time ... this unnecessarily places a burden of stress on you and those around you. We need to manage time, not have time manage us.

JOB ENVIRONMENT

Work claims a large portion of our time, labels us with titles and self images, dictates where we live, how well we live, how often we move or travel, and what time we get up in the morning. It is a major source of stressors in our life, and also a major starting point for reducing stress.

PHYSICAL WELL-BEING

Your body's reactions to stressors become stressors in their own right. Illness, pain, obesity, chronic substance abuse, all take their toll on your body and on your stress resistance.

The physical symptoms are red flags that warn you that stress is having an impact upon you. It is your body's way of telling you to slow down and take a closer look at what is happening in your life.

DEALING WITH STRESS

So, why do some people become buried under stress while others in similar situations appear unaffected by it?

The stress response is a response to change. The change may be positive, such as an exciting vacation, or negative, such as the death of a spouse.

The key word here, however, is "response". Each of us evaluates a potentially stressful situation differently. The way in which an event is perceived is often more important than the objective nature of the event. In other words, one person's stress is another person's exciting experience.

Our response to stressors results from a complex framework of factors, such as:

>The amount of control we have over the stressor

>Our resources for dealing with the stressor

>The duration of the stressor

>What other stressors happen concurrently

>Our physical condition at the time

>How we learned to deal with stressors in the past

>How threatening the stressor is to us

Three methods of dealing with stress have resulted from studies of this framework. One way is to determine the factors that can trigger stress and find ways to reduce or avoid them. Another way is to alter your mental attitude toward the factor so that it does not trigger stress in the first place. A third way is to practice techniques that make us "stress-resistant" and better able to cope with stress if it comes our way.

AVOIDING STRESSORS

CONTROL

An important issue in stress reduction is the amount of control you have -- or think you have -- over your environment.

Studies have shown that middle managers have more stress-related problems than top executives faced with similar problems. This is because the top executives have the power to change the environment in order to solve a problem, while the middle managers must deal through the structure of the organization with much less power and authority.

Even rats, given random electric shocks, survive with fewer stress symptoms when given the ability to control the duration of the shock.

There are a number of things that you can change in your life to increase the control you have over your situation:

>Do not accept responsibility without the necessary authority to follow through.

>Set clear-cut, reasonable goals for yourself and get agreement on them from those who will be involved.

>Follow through on your goals.

>Make your own expectations very clear.

>Give feedback to others when accepting responsibility or communicating information. Make sure that there is a consensus on what is being communicated.

>Lobby for personal control over your work environment. Flex time is a good example of this, and it has been proven to increase productivity and reduce stress among employees.

TIME

Each day has 24 hours ... no more, no less, and the need for more time to do what we need to do is a constant source of stress for most people. Since we cannot create more time, we need to develop techniques for better management of the time that we are given.

Time management is a complete subject unto itself, but a few tips on increasing the effective use of your time follow:

>Block out interruptions when working on a task.

>Learn to say "NO", gracefully and firmly.

>Find major time-wasters, such as waiting in lines, unplanned interruptions, errands. Find ways to reduce or eliminate them.

>Schedule "private time" for rest and reflection. You will be more effective if you are relaxed.

>Put things where they belong.

YOUR BODY

Lack of sleep, improper diet, and lack of exercise all contribute to ill health. An unhealthy body is a body in stress. You have choices to make every day about your physical health, and there is a wealth of information available to guide you. Become aware of those choices and cater to your body's needs.

YOUR SURROUNDINGS

Living and working in urban America, we are surrounded by noise, air pollution, heavy traffic, crowds of people, and allergens. These things have become a fact of life for many of us, but there are ways to avoid them or reduce their effects. Here are a few:

>Use flex-time to change your commute to a less crowded time slot.

>Live close to your work

>Insist on clean, well-lit, low-noise working space.

>Find ways to adjust your work space to be more ergonomic.

THE MEDIA

What we read, watch, and hear all have a profound effect on our level of stress.

Bad news seems to be a fact of life in this age of electronic information gathering, but we have a choice about how and when we hear it. Make it a habit to read more deeply than just the headlines and TV newscasts. Tape the news and skip over the sensational parts. Be selective about what you watch and watch at your convenience, not just at 6 and 10 o'clock.

ALTERING YOUR ATTITUDE

Why do some people respond with more obvious stress symptoms than others when confronted with similar stressors? Studies have been done comparing low- and high-stress individuals, with the conclusion that attitude and interpretation have more to do with the stress level than the actual stressor does.

For example, a study of Norwegian parachutists shows that the first jump is the most stressful. After the action becomes familiar, it is no longer perceived as a threat. Same action ... reduced stress.

Since the stress reaction is difficult to control once it is set into motion, initial evaluation of a situation is crucial to the body's management of stress. The challenge is to find ways to channel our perception of situations toward a positive, enabling viewpoint.

REFRAMING

Most situations can be viewed from two sides. A move to a new community will threaten a family with the loss of old friends and the need to adjust to new surroundings. The move itself is a major hassle of packing and unpacking. On the other hand, the family will make new friends and the old friends will certainly not be forgotten. Finding new doctors, shopping centers, and schools can be exciting and will broaden the experience of the entire family. Packing up your worldly goods gives you the opportunity to remove unnecessary possessions and to find lost objects. Looking at a situation realistically and optimistically gives you a psychological "escape route" from stress.

COMMITMENT

Having a commitment to some overriding goal or belief has been found to be a major factor in coping with stress. Commitment gives you a focus for your actions and decisions.

UNPRODUCTIVE THOUGHT PATTERNS

We've all at one time or another responded to unproductive thought patterns. These are patterns that were developed earlier in life to deal with the threats at that time. If a pattern seems to work, your mind tries it again. Soon, you use the pattern automatically when a similar threat is perceived. Over a lifetime you can build up quite a repertoire.

In stress situations, we tend to fall back on familiar patterns. This type of behavior causes you to act out of congruence with the situation, and adds to the stress.

Learn to recognize these thought patterns for what they are and re-evaluate them in the light of what is currently happening. Listen to your internal monologue. Is it appropriate to the situation? If not, consciously reject the thought. Pretty soon, the response will no longer be on the "automatic" list.

BECOMING STRESS-RESISTANT

There are many techniques that can be used to increase our coping skills, some that can be practiced daily to build stress-resistance, and some that are handy to use when stress actually arises. A few of these techniques follow:

>Circular breathing

>Focused effort

>Mini relaxation breaks

>Body scan

>Exercise

>Meditation

>Laughter

These techniques take some effort on your part to initiate, but the payback is great compared to the effort. Most can be included easily in your daily schedule, and will soon become unconscious habits if used regularly.

CONCLUSION

In order to get a handle on stress in your life, you will need to make some tough decisions. Your lifestyle, your way of thinking, your habits, your surroundings, all are potential sources of stress ... or opportunities for stress reduction. Develop more appropriate patterns for yourself, and influence others around you to do the same. Remember that positive attitudes are contagious.

BIBLIOGRAPHY

.Brunner, Margaret, "Take A Deep Breath" (1989).

.Faelton, Sharon; David Diamond, and the Editors of Prevention Magazine, Take Control Of Your Life, A Complete Guide To Stress Relief (Pennsylvania: Rodale Press, Inc., 1988).

.Frecknall, Peter, Ph.D., "How Much Can Mind-Body Techniques Save by Reducing Absenteeism?" in the Journal of Mind-Body Health Advances (Spring 1989 Vol 6. No. 1).

.Gazzaniga, Michael S., Mind Matters (Boston:Houghton Mifflin Company, 1988).

.Perkins, John M., The Stress Free Habit, Powerful Techniques for Health and Longevity from the Andes, Yucatan and Far East (Vermont: Healing Arts Press, 1989)

.Rosen, Robert, "Healthy People, Healthy Companies: Striking the Critical Balance" in the Journal of Mind-Body Health Advances (Spring 1989 Vol 6. No. 1).

.Withers, Pam Miller, "Good Stress, Bad Stress" from Working Woman (September 1989)

.Watson, Rita Esposito, "Physical Symptoms" from Working Woman (September, 1989)

5156

UNATTENDED DATA CENTERS..... FANTASY OR REALITY?

Frank J. Pinkela
Unison Software, Inc.
675 Almanor Avenue
Sunnyvale, CA 94086
408-245-3000

ABSTRACT

Today's modern data center, while boasting the latest in hardware technology, continues to struggle with the implementation of software that can automate the operational daily production activities. While much has been written on "unattended" operations and "lights out" data centers how much progress has really been made in actually accomplishing these objectives?

Does top management understand the cost trade-offs associated with the data center in terms of personnel turnover, production reruns, policies and procedures regarding production schedules, user submitted jobs, backups, restores, report distribution, and console messages. How will today's data center manager deal with managements request for more productivity and less people? What will be the affect on the data center with the increased use of UNIX boxes? Is it possible to continue to increase these "operatorless" tasks through software?

This presentation will focus on industry trends and case studies in the move toward more automated data center activities. Included will be data from both HP and IBM data centers as well as information from vendors and consultants.

This talk is directed toward supervisors and managers of data center operations for all HP platforms, MPE V, MPE XL, and HP-UX.

BACKGROUND

A recent gee-whiz article appeared in an AFCOM publication (Dec. 1990) stating - "The National Commission on Superconductivity recently recommended developing a super computer operating at "petaflops" speed, about one-million times faster than today's fastest computer. By contrast, it is estimated that the human brain computes in the 10 "teraops" range, faster by a factor of 10,000 than today's best super computer."

Figure 1 shows the current offerings from HP in terms of relative performance of the HPPA machines. Please note that the current upper end machine operates at about 50 MIPS and costs about $1 million. That makes you (or more specifically your brain) worth, on the above scale roughly $20 billion. Amazing!!

HPPA

Relative Performance Chart

	920	922LX 922RX 922	932	948	958	955	960	980/ 100	980/ 200
Relative Performance	**1**	**1.7**	**2.8**	**6**	**8.5**	**5.4**	**7.5**	**13**	**20**
US List (preconfigured)	$26K*	$35K* $65K* $75K*	$89.5K (<64 users) $99.5K (>64 users)	$160K (<64 users) $190K (>64 users)	$310K*	$385K	$485K	$675K	$1050 K
Max. logged on users	20	32 64 152	64 or 152	64 or 152	600	600	600	850/ 1250**	850/ 1250**
Standard/Max. Memory (Mb)	24/56	24/128 32/128 32/128	32/128	64/256	96/256	96/256	128 /256	192 /512	256 /1024
Max. Disc (Gb)	12	12	22	34.8	34.8	85	85	85	85
Available	Now	Now	Now	12/90	12/90	Now	Now	Now	1Q91

* Base price includes integrated disc and tape

** Available 2H91

This anecdotal reference clearly points to continued advancement in the computing power of hardware as we progress toward more and more efficient ways to conduct process data in both our personal and business environments.

The objective function in automating the data center should be in maximizing efficiencies and minimizing costs. Look at what has happened in your own company's business operations, automated production lines, robotics, automated warehousing and inventories, automated cash registers, point-of-sales equipment, CASE tools for development, computerized-digitized everything. Has automation of the computer data center kept pace with these advancements? Part of the problem, I think, is the failure of the "keepers" of the automation processes to use these tools in their own environment. Kind of like the old bromides about the shoemaker's son going barefoot or the doctor unable to deal with his own prescription for good health.

SMR STUDY RESULTS - by Sentry Market Research, Westboro, Mass.
These study results address a wide range of automation issues and measures the current impact of automation at a wide variety of computer sites. Typical respondents were involved in technical operations & support and systems administration.

Research shows that AO tools are delivering results on target with customer expectations and further suggest that employee morale has NOT been negatively affected by the automation of the data center.

A sampling of the group showed they made a significant investment in time in documenting the "before and after" effects of the automation process.

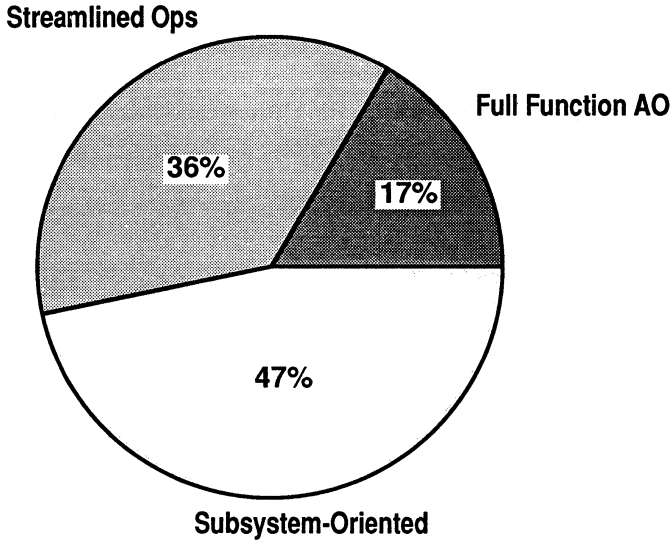
Cost savings were tops on the justification list.

Head count reduction was the least important item on the list.

While the market place abounds with automation tools, most vendor products fall short of user expectations for interchangeability. A study I conducted last year with 10 HP data center managers indicated they were looking for more "single-vendor" solutions and a move toward "open" exchange of information, common operator interfaces, and alliances amongst vendors to standardize automation approaches.

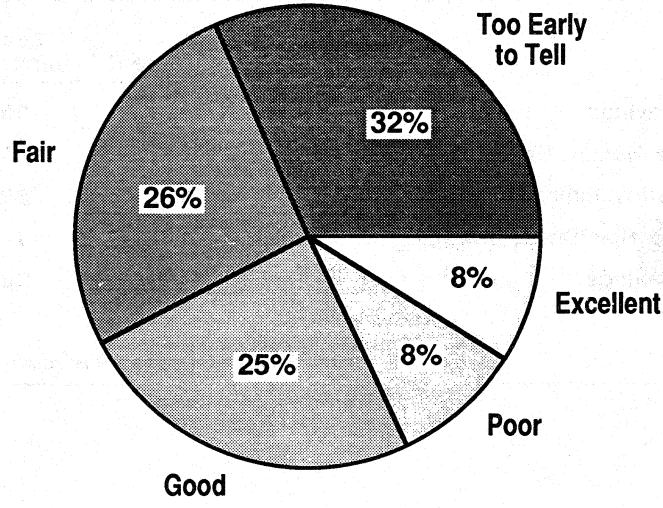
Relatively few companies have "lights out" as their ultimate goal. This is as much a testimony to the current state of AO technology as it is of the relative uneasiness users have with totally unattended operations. The level of comfort seems to be around more of a "dim the lights" rather than "lights out".

Current Level of Automation in Data Center Operations



Source: Sentry Market Research 1990

User Success Ratings with Automation Technology



Source: Sentry Market Research 1990

Automated Operations Software

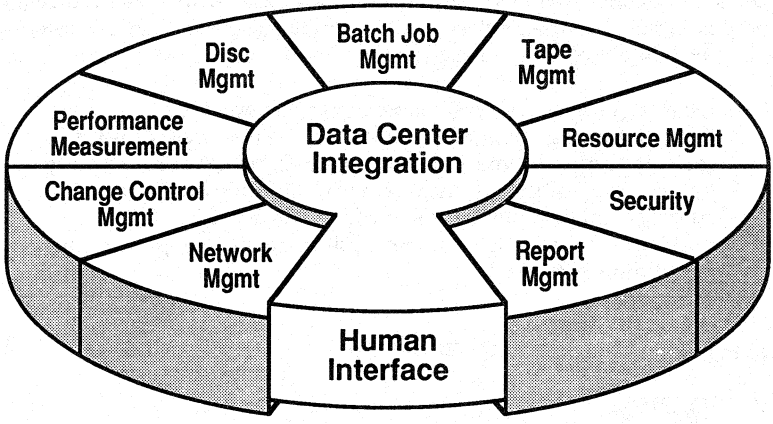
Purchase Justification

	Most Important	Indifferent	Least Important
Cost Savings	53%	22%	25%
Reduce Human Error	41	31	28
Availability Improvement	40	31	29
Improve Hdw/Sfw Efficiency	35	31	34
Staff Savings	26	36	38

Source: Sentry Market Research 1990

Automated Operations Software

Open Architecture for Future Development



OPEN ARCHITECTURE FOR DATA CENTERS
(Ref Figure 11, AUTOMATED OPERATIONS SOFTWARE PIE CHART)

CHALLENGES FOR DATA CENTER MANAGERS

A. CRITICAL SUCCESS FACTORS

Managers need to have a handle on "critical success factors" in order to establish a baseline operational position and to measure the effects on additional workloads. Management is now asking, and will continue to ask the question, "How can we do more with less resources?" Are you prepared to answer , objectively?

The term "critical success factors" is used to describe the various production activities and performance factors that highlight the productivity and efficiency of a data center needed to communicate with top management as to operational performance data...how are current resources being used; why are additional resources needed; and, what is the benefit to the enterprise if the capital is directed to this project; are there better projects than this one for this capital; etc. Some of these measurements might be,

-resource usage - cpu, disk, tape, network, supplies.
-user computing - planned, unplanned, peaks, need vs want.
- on-line response time - demand periods, perception vs reality.
-system availability - downtime, weak link in availability.
-batch job throughput - number of jobs processed.
-problem analysis - job failures, system downtime, prod reruns, scheduling bottlenecks.

WHEN TO AUTOMATE?
(Ref Figure 14...When to AUTOMATE...)

CASE STUDIES...
(Reference Figure 23...Case Studies)

A. COMPUTER EQUIPMENT MANUFACTURER

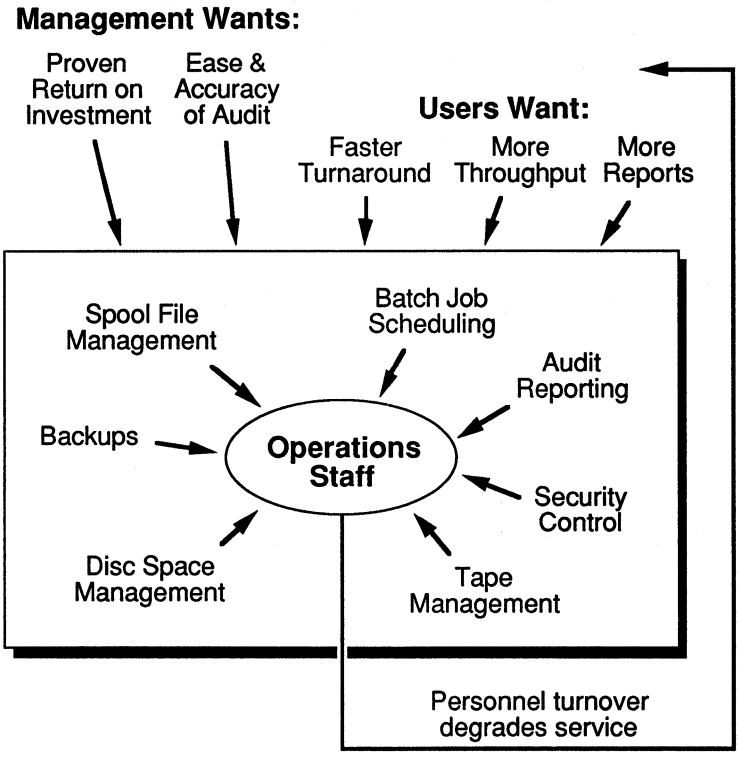
B. COMPUTER PERIPHERAL MANUFACTURER

C. FINANCIAL SERVICES COMPANY

When to Automate?

- Rising staffing costs and workloads
- Demand for higher throughput
- More complicated applications
- Increased efficiency with less cost

Pressures on the Data Center



Data Center Management

Automation

Why?

Cost Savings
Staff Reduction
Improved Efficiencies

How?

Any way you can

When?

DO IT NOW !!!

Data Center Management

- Hardware/Data Center Reliability
- Change Control
- Problem Tracking
- Separation of Duties
- Automation
- Standardization
- Security

Data Center Management

System Management

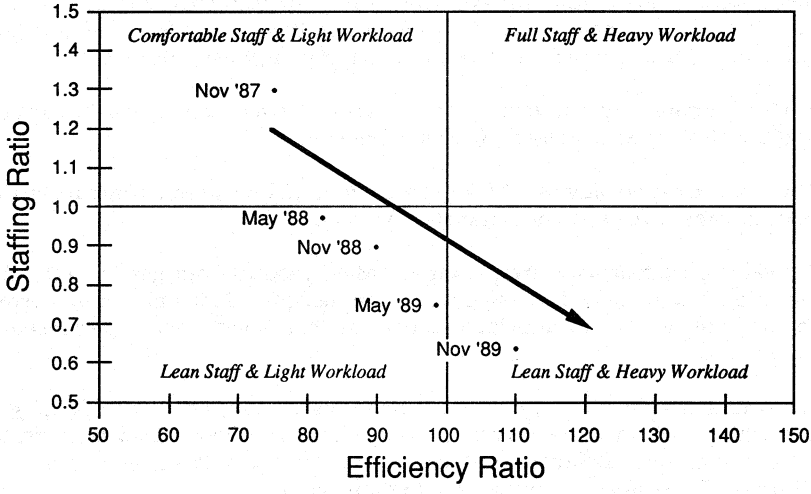
- JCL's
- Job Scheduling
- Capacity Planning
- Performance Monitoring
- System Recoveries
- Backups
- Tape Management
- Report Distribution

Data Center Management

Production Scheduling

- Increased throughput
- Elimination of reruns
- More effective job scheduling
- More effective use of resources
- Reduced need for operations staff

Data Center Productivity
Staffing Ratio



SUMMARY

Yes, unattended operations are a "reality."

Many are moving into more and more automation of the data center for reasons of economy, available resources, ability to respond to situations, and the need to do more. Some have arrived and are fully committed to this automation concept, others are making do, and still there are a few that have no interest.

If you're in any of the above, except for the last category, then the following may be useful to your quest for automated operations:

1. Evaluate your in-house software for scheduling, restart, backups, report distribution, and console management. Are your packages working together or is it feasible to make them work together in the future. ESA.....if you can't eliminate it, then simplify it; if you can't simplify it; then automate.

2. Differentiate between your current costs and what will it cost to support additional planned workload. (baseline budget).

3. Consider expert advise. Make use of your fellow users, companies that have advanced the state of automated operations.

4. Evaluate current software packages. Most products can get the job done. Question - how easy and effective to implement. Not only your current "pains", but those your anticipate and those that others have experienced in the process.

5. Document your plan. Plans alone don't mean a lot, however, *planning* is everything. Identify the costs and savings that will be realized through the purchase of new software and track your progress (both successes and failures.) **Remember, you may have to do it again.**

While the cost of upgrading can be expensive, the payoff will be efficiently run and competitive data centers.

$pending to $ave on $oftware

presented by

Shelby Robert
General Manager
Proactive Systems
Four Main Street, Suite 101
Los Altos, CA 94022
(415)949-9100

A. Introduction

As this paper is written, things look pretty grim for the economy. Many organizations are facing budget reductions and cutbacks in spending that are unprecedented in the memory of an entire generation of MIS people. Times are tough and so are the managers who control the pursestrings.

Still there are times when a purchase of software really will save the company real dollars in terms of reduced costs, or in terms of increased productivity. So the question facing the MIS manager becomes "How can I demonstrate to my management that spending a few dollars now will help my company to do better in these troubled times?"

This is the question that must be answered after you have completed a thorough technical evaluation. The product has been selected. You're satisfied that you have the *right stuff* in hand, but now it has to be justified to top management. It is necessary to convince them that the decision to purchase now is the best thing that the company can do to move itself forward.

In the end this paper is about developing numbers that support your decision to move forward with a purchase. Now we approach a different task. The job is measuring the relative value of making this purchase as opposed to not making it. How positive will be the effect on the company of making this purchase?

B. The VALUE Equation

The real question to be answered in looking at acquiring software is *What value will my company derive out of this purchase?* If the answer is nothing, then we might as well stop right here. If, on the other hand, you perceive that there is real value for you and your company, the balance of this paper is designed to help you measure that value.

So how do you measure value? One definition that seems to hold up pretty well is:

VALUE = BENEFITS - COST

We will refer to this as the "Value Equation." It is obvious that as measured benefits rise, and as the costs decline, the value of the purchase to your company increases. The greater the difference you can show between the benefits of acquiring an item and its cost, the more convincing and persuasive your presentations will be. Because the price of your proposed acquisition is fairly obvious at this stage, the real key to understanding the power of the Value Equation is in measuring benefits.

C. Identifying Benefit Areas

Measurable dollar benefits for the MIS department generally come in one of four basic varieties:

a. Reduction of materials costs
b. Increased MIS Staff productivity
c. Increased User Staff productivity
d. Reduced space/power/support services consumption

Individual software purchases will not necessarily find benefits in each of these areas, but most of the major measurable benefits will fall into one of them. At this point, an example may help to illustrate these issues.

One of the issues facing many MIS departments today is the question of replacing preprinted forms with electronic ones. Let's examine some benefits of this way of

doing business that touch on each of the four areas mentioned above.

So that we are on some common ground, electronic forms are forms that are printed at the same time as your data on a LaserJet printer. The obvious advantage is that there is no need to have a supply of preprinted forms on hand in order to print such things as invoices, purchase orders, packing lists, and other documents. They simply print on plain white paper, including your logo, with terms and conditions on the back, at the selected printer, and with the appropriate number of copies. All of this sounds great, but what are the measurable benefits here?

Benefit #1. Cost of the forms.

The first and most obvious benefit is that the cost of the preprinted form goes away. But this paper is about measuring things, so let's play with some numbers. Suppose you run a manufacturing company's MIS department and each working day you print 500 sets of four-part invoices that cost you $0.20 per set. At 250 working days per year, your annual consumption is 125,000 for a total annual cost of $25,000.

Unfortunately, there are some materials costs associated with LaserJet printing which must also be factored in here. The toner cost per face is about $0.015, and since our invoice requires terms and conditions on the back of the original, we will have five faces to print. In addition, the paper itself costs about $0.004 cents per sheet, and we need four sheets. So our total costs to produce the same form as above are:

125,000 * (($0.004 * 4) + ($0.015 * 5)) = $11,375.00

To determine the real measurable benefit to the company, we now subtract the old costs without our software acquisition from the projected new costs with the software in place to find the dollar amount of the benefit.

$25,000.00 - $11,375.00 = **$13,625.00 per year**

So far in examining this benefit we have looked at replacing only a single form. If you were actually contemplating such a purchase you would list the savings to the company for each of them. The formulas remain the same, only the names are changed to protect the expensive. So let's explore one more just for good measure.

Another form that you probably deal with is a monthly statement. Let's assume that our fictional company has 1,000 active customers, and that the statement is in two parts, one for the customer, one for the Accounts Receivable department. Using the

same numbers as above, the preprinted set will cost about $0.10 per set. Printing 12,000 per year results in an annual paper cost of $1,200.00.

Again using the same costs as above, two sheets of paper at $0.004 each and three printed faces with a toner cost of $0.015 per face yields a set cost of $0.053 per set. At an annual cost of $636.00, this is still a savings of $0.047 per set or $564.00 over the 12,000 printed in a year.

$1,200.00 - $636.00 = **$564.00 per year**

Just a word here about presentation. All of us like to have information about which we are being asked to make decisions presented in an easy to follow, understandable way. Although the supporting details are important to reveal thoroughness in the discovery process, the major points are the ones that need to be highlighted. It is nice for the reviewer to have available all of the details, but he will appreciate a presentation that helps the important numbers jump out so that his time and energy are saved the task of extricating them from the "Spreadsheet Sea." One of the ways to improve presentations is to tabulate results.

If you have several forms to deal with, you might want to summarize in a format that looks something like this:

Form Name	Sheets per Set	Savings per Set	Annual Usage	Annual Savings
Invoices	4	$0.109	125,000	$13,625.00
Statements	2	$0.047	12,000	$564.00
			Total Annual Savings for Benefit #1 =-->	**$14,189.00**

Benefit #2. MIS Staff Overhead Reductions

The MIS staff have a built-in burden in handling pre-printed forms. When forms printing is needed, the forms must be mounted on the printer, printed, decollated, burst, and distributed. This is apart from a weekly inventory, and the involved process of reordering when quantities run low. Consider the following numbers for our sample manufacturing company:

Operations time (daily):

a. Printing forms	1 hour
b. Decollate / Burst	1 hour
c. Distribution	1 hour

A total of three hours per day is 15 hours per week times 52 weeks totals to 780 hours per year. Plus additional overhead of one hour per week for the weekly inventory. This leaves us with a total MIS operations overhead of 832 hours. There is a lot of training and investment in MIS operations people. The cost rate for them is relatively high in comparison to office or plant workers. A rate of $50.00 is not unreasonable in figuring total costs for the skill level and responsibility required of these folks.

All of the above overhead in the MIS department can go away entirely with electronic forms, so all of the costs are either direct savings, or opportunities for MIS staff to be productive doing other things. Either way the benefit to the company is:

832 hours * $50.00 = $41,600.00

Every time a form must be reordered, it is reviewed by management. If not only by MIS management, then by other managers in the company. The review must be painstaking because reorder points are infrequent and volume purchases, necessary to get the unit price reasonable, are still quite expensive. To get away with reviewing a form in four hours seems quite reasonable. The people who must do the final review are sophisticated management people. Their hourly cost is reasonably in the $100.00 per hour range. So with 20 forms to review per year our current cost equation is:

20 forms * 4 hours per form * $100.00 per hour = $8,000.00 per year

Because electronic forms can be changed on a moment's notice, lengthy regular periodic reviews are not required. Some overhead will still be necessary, but reductions of 75% are not unusual. This still leaves us with a cost savings of $6,000.00.

The total of reductions in MIS overhead is now the combination of the two numbers from above.

Benefit #2 = $41,600.00 + $6,000.00 = $47,600.00

Benefit #3. Increase in User Productivity

No MIS department survives in a vacuum. MIS embodies the very definition of a services business. We have no reason to exist without our users. To go one step further, it is our users who are doing the business of the company, not those of us in MIS. If we can contribute to the productivity of our users, we will have a direct beneficial effect on the company's ability to do profitable business. One of the ways that we do that is by reducing the time it takes us to deliver information to our users.

In using pre-printed forms, we wait and batch things like invoices until we have an opportunity to mount them on the printer. If the picking list is the fourth copy of that invoice, the goods can't go out to the customer until the invoice is printed. The third copy of the invoice must be carried to accounting for their needs, and the original and remittance advice will go to the mail room to be posted.

With the introduction of low cost LaserJet printers and the availability of software to drive them, we can now print the original and remittance advice in the mail room, the picking list in the warehouse where it is needed, and the accounting copy in the accounting department, all without any delays. The overall result is that needed information arrives at its destination where it can be used in a more timely fashion. Productivity will rise due to increased efficiency of delivery of information.

To illustrate how powerful productivity gains can be, let's assume that our manufacturing company has five people in accounting and ten in the warehouse. If we can realize a modest 1% productivity increase for those people by improving information delivery, we will save 6 hours of labor per week that can be used for accomplishing other tasks.

At $20.00 per hour average cost to the company to have an employee on staff and working, the annual productivity gain to the company is:

Benefit #3 = $20.00 per hour * 6 hours per week * 52 weeks = **$6,240.00**

Benefit #4. Reduction of Space

The 20 forms that we have been using must be individually stored and inventoried each week. They use floor space in the computer room for the working store. In the storage area which is secured, they use additional space.

Computer room space is expensive at about $10.00/sqft per month. At only one

square foot per form, the cost is $200.00 per month, just to store forms. In the storage area the cost of $3.00/sqft per month is less, but each form will require an average of 10 square feet. This amounts to $30.00 per form per month or $600.00 per month in the storage area.

Again, the total will not be reclaimed, but at 75% savings (because we now will stock only white paper) we will reclaim $600.00 per month of floorspace. The annual total is:

Benefit #4 = $600.00 per month * 12 months = $7,200.00

D. The Bottom Line

The benefits listed above are not an exhaustive list by any stretch, but they serve to illustrate how to identify and measure the benefits that will accrue to your company by making such an investment. Now we can put it all together for your management.

First, a brief review of the benefits that we have identified and their dollar value to the company.

Benefit #1. Direct Forms Cost Savings $14,189.00

Benefit #2. Increased MIS Productivity $47,600.00

Benefit #3. Increased User Productivity $6,240.00

Benefit #4. Reduced Space Requirements $7,200.00

Total Annual Benefits $75,229.00

All of this does assume the purchase of a software package to do all of these marvelous things, and the purchase of some LaserJet printers to do the work. The discussions above have mentioned a printer in the warehouse, a printer in the accounting department, and one in the mailroom. MIS will of course have to have a printer to do their testing. Assuming that all of the printers purchased were LaserJet IIIDs, the total investment in printers would be about $8,000.00. The software to do the job above will cost an average company about $7,000.00.

These two numbers represent the total capital investment for the job. The problem is that comparing the $15,000.00 total of these two investments to our benefits list is like comparing apples and oranges, since our benefits are expressed as annual

amounts, but the benefit from the investments will last longer than one year.

The best way to evaluate the viability of a capital investment is to use a discounted cash flow methodology. A discussion of this method is a little beyond the scope of this paper. Another, simpler approach is to look at the amortized cost over the life of the investment and the payback period.

Capital investments must be amortized over their "useful life." The tax man defines useful life for computing investments as being five years. What this means is that only one fifth of the cost of the hardware and software should be taken as an expense in any single year. In our example, this represents an annual capital cost for our project of only $3,000.00.

Notice also that the benefit numbers above are all annual numbers. That means that these benefits will accrue to the company each year of use. So now a look at our bottom line numbers:

Total Annual Benefits **$74,665.00**

Total Annual Capital Cost **$3,000**

All of the things we have mentioned above are important. They are the essence of what must be communicated to your management, but so far they are only numbers. For many people, numbers don't have a life of their own. They just sit there. In reality, numbers by themselves are just plain

```
Boring!!!
```

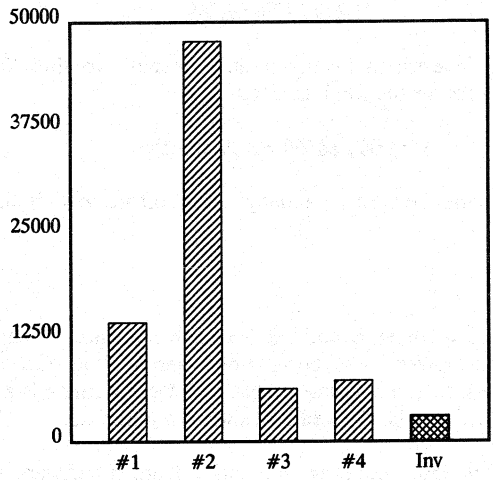
For you as an MIS Manager, this means that you have to do something to get your management's

ATTENTION!!!

How can you do that? If the problem is that all you have to work with is numbers, why not paint your management a picture of what the numbers mean. Pictures of numbers in graphic form convey meaning instantly, intuitively, without having to do the mental work of internalizing the specific value of each benefit. By painting management a picture, you save time, help them to make a better (as well as faster) decision, and make it easier for them to accept your arguments in the process. In the process you don't do any harm to your own reputation for professionalism.

What might a picture of our results look like? Let's take a look at the effect of illustrating the relative value of each of our benefits and our annual investment.

Individual Benefits vs. Investment
Annual Amounts



By looking at the above chart, it is obvious immediately that any one of our four benefits would be sufficient to justify the purchase. The combination of the four is an obvious and overwhelming argument.

Let's go back now and take another look at the VALUE EQUATION. The VALUE EQUATION above stated that VALUE = BENEFITS - COSTS. Solving the equation with the numbers we have developed here yields an ANNUAL VALUE to the company of $69,865.00. Over the five year life the tax man requires, this investment will return $349,325.00. Even the most callous of CFOs will take note of a return like that.

E. Time to Return

One of the calculations that you will want to make has to do with the time to return the investment you are asking your company to make. The easiest way to do this is on a monthly basis. First, divide the annual benefit by 12 to find the monthly benefit.

$75,229 / 12 = $6,269

Now divide the total investment by the monthly benefit to find the number of months it will take to recover the total invested.

$15,000 / $6269 = 2.39 months

Again a very favorable number. This company cannot afford to wait another day for this purchase.

F. Summary

In the preceding pages we have examined only one software purchase, that of a LaserJet printing package. We have certainly not examined all of the benefits that might be associated with such a purchase. What we have examined is a methodology for quantifying the benefits to be derived by a software purchase.

The basic benefits that your company will derive from a software purchase will generally come from one of the four sources mentioned above: reduced materials cost, increased MIS staff productivity, increased user staff productivity, or reduced consumption of other company resources. In order to justify such a purchase to management, the individual benefits need to be itemized, and quantified into dollar savings or productivity gains.

Once the benefits are identified, quantified, and valued, a little care and attention to the presentation of the information goes a long way with management. If you are convinced that this is the right decision for your company, it is worth spending a little time to make it easy for your management to understand your position.

The numbers that you generate in preparing a cost justification may yield totals that seem unreal. Your benefit amount may be dismissed as not being a hard dollar savings. The reason is that your company is probably not going to fire anyone to reduce their costs as a direct result of this sofware purchase. Nevertheless, the arguments for the benefits still hold because the time saved can be put into more productive work.

The important issue in justifying software is that the benefits to your company must be itemized and valued in order to demonstrate to your management that the acquisition you propose is the right thing for the company to do, and that this is the right time to take action. By going through the exercise of identifying and quantifying the benefits of your purchases, you will strengthen your bargaining position and your case for the purchases that you need to make to help your company function better.

This article was typeset using FANTASIA from Proactive Systems.

Popular Mass Storage: Optical Disk and Helical Scan Tape

Husni Sayed
IEM, Inc.
P.O. Box 8915
Fort Collins, CO 80525
(303) 223-6071

Introduction

Optical disk drives and Helical-scan tape drives, relatively recent advances in the mass storage arena, are transforming mass storage technology. These two technologies can offer significant advantages over more traditional storage methods, and each has its own specific advantages.

Helical-scan tape is very inexpensive, and has a large enough capacity to allow unattended backup of on-line systems. It would take almost 13 reels of high density (6250 bpi) 9-track tape, or 12 IBM 3284 cartridges, to store as much information as a single 8mm tape cartridge. Furthermore, the traditional tape system would require the presence of an operator to change every one of those reels or cartridges: a process that may take hours. This additional labor cost makes helical-scan tape even more attractive. Decreased storage space is another incentive that favors helical-scan tape.

Optical disks have many of the advantages of helical-scan tape, such as large capacities in small volume, and also offer faster access times. They are serving to fill a previously empty niche that existed between large capacity, low cost mass storage tape systems, and fast access, high cost Winchester disks. The removable nature of optical media makes it ideal for storing large software systems and data bases that can be swapped in and out as needed.

Optical Disk Technology

Optical recording devices, which use lasers to store and retrieve data from optical disks, were first developed as an alternative to the Video Cassette Recorder. In 1978, the first optical disk system — the Laser-Disk Video Player — appeared on the consumer market. This read-only device used a 12" platter and a laser read-head to play back digitally encoded video signals.

Since then, optical recording technology has been further developed by the mass storage industry, and split into three distinct branches: CD-ROM (Compact Disk-Read Only Memory), WORM (Write Once, Read Many) and Rewritable (Erasable).

The main expense involved in optical disk systems is the read/write head, which uses lasers, beam-splitters, lenses, and mirrors to access data. For this reason, most optical systems are single-sided: the disk must be removed from the drive and manually turned over to access the other side of the disk cartridge. An optical disk system that could access both sides of a disk cartridge without removing the disk would require two read/write heads, effectively doubling the cost of the drive.

Optical Disk Advantages

Optical disks have numerous advantages over magnetic media. First, since the density of an optical disk cartridge is limited only by the wavelength of light used by the laser writing the information, optical disks are capable of tremendous track capacities. Most current systems use a near-infrared laser with a wavelength of 8,000 – 10,000 angstroms, resulting in track densities on the order of 16,000 tpi. A single $5\frac{1}{4}$" optical disk cartridge with such a track density can hold roughly 800 MBytes of data. Second, the distance between the head and the surface of the disk is much greater than that used by traditional Winchester technologies. This increased separation between the disk and the head makes head crashes very rare. Finally, the optical disk cartridge itself is very durable. Encased in plastic, it is immune to fingerprints and resistant to heat and humidity. These factors combine to offer an archival life of more than 10 years, in a disk that is removable and easily transported from one machine to another.

However, optical disk drives have slower access times than Winchester disk systems (50–150 ms as opposed to 10–20 ms). This is because the head in an optical disk drive weighs much more than the head in a Winchester. Data transfer times, which are dependent solely upon rotational speed, are comparable between the two systems. However, in applications where speed is crucial, the optical disk drive may well be outperformed by magnetic hard disk.

CD-ROM (Compact Disk-Read Only Memory)

Characteristics

CD-ROM (Compact Disk Read Only Memory) disks are high capacity "Read Only" memories: information can be written to the disk only during the manufacturing process, not by an end-user. This characteristic limits their usefulness as a typical mass storage medium. The disks themselves are thin, flat disks made of polycarbonate covered by a thin reflective layer. Each disk is roughly 4.75 inches in diameter, and just over a millimeter thick. The medium used for computer applications is physically the same as that used for the music industry, so anyone who has been in a music store has seen a CD-ROM disk.

Information is stored on only a single side of the disk, with capacities of more than 500 MBytes per disk.

Implementation

The surface of a CD-ROM disk has a continuous spiral track, like a phonograph record, but with a much higher density. A "pit" on the raised portion of the track represents a digital 1, and a flat area (or "land") represents a digital 0. A master disk is used to duplicate the information by "stamping" the information onto other disks.

The stored data is read using a low-power laser: pits and lands on the surface of the disk reflect light differently. This difference in reflective quality is detected, and translated into readable data.

Applications

Because the process of preparing data and creating a master disk is so expensive, CD-ROM is not economical unless a great many copies are going to be made. For this reason, CD-ROMs are largely used to distribute and reference large amounts of relatively static data such as on-line encyclopedias, legal citations, and (of course) musical recordings.

WORM (Write Once, Read Many)

Characteristics

WORM (Write Once, Read Many) optical disk are more useful as a mass storage medium for the typical end-user. Unlike CD-ROMs, information can be written to the disk by the end-user—however, information can only be written once, as the writing process causes permanent alteration of the disk surface.

WORM Optical disk cartridges typically come in sizes of 5.25 inches and 12 inches, with the 5.25-inch size more popular. A WORM optical cartridge, unlike a CD-ROM disk, is actually encased in a plastic casing. Also in contrast to CD-ROM disks, information can be written to both sides of the disk, although the disk must be physically turned over to access information on the second side. Capacities on a WORM disk are typically 600 to 800 MBytes (300–400 MBytes per side).

Implementation

Like CD-ROM disks, WORM optical disk drives write information using a laser which burns pits into raised portions of a spiral track on the surface of the disk. Once a pit has been created, that area of the disk cannot be restored to its normal flat surface: thus information written to a WORM disk is permanent. Figure 1 shows a vertical cross section (along tracks and sectors) of a WORM disk.

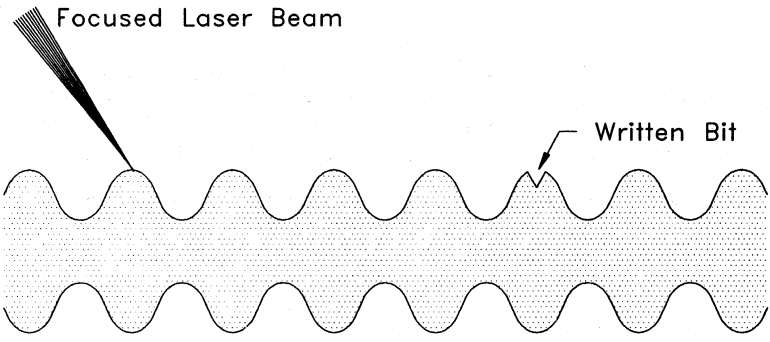


Figure 1: WORM Optical Recording

To read the data, the same laser is directed at the surface, but at a much lower power setting. The laser is reflected off the surface, and this reflected light is gathered into a photocell. The light reflected by a pit is easily distinguished from light reflected by a flat surface: this difference in light reflections is used to read bit patterns. Each pit is interpreted as a digital 1, and each land ("no pit") is interpreted as a digital 0.

Applications

WORM optical drives have one major "snag" that is not encountered with other mass storage technologies. Most existing file systems are structured so that some space on the disk is reserved for a directory. This directory must be updated each time a file is added, edited, or deleted. Since WORM optical disks cannot be rewritten, such directory maintenance is impossible. One solution to this problem is to employ special software drivers that use an entirely different file structure involving linked directories. Other solutions involve using a flexible disk to store the directory entries, while the actual data is stored on the optical disk. Directory maintenance limitations, combined with the write-once nature of the media, make WORM optical disk drives useful primarily for backup and archival tasks.

The write-once "limitation" of the medium, however, gives the WORM optical disk drive one unique advantage: once data is written, it cannot be altered. This characteristic makes WORM drives excellent for storing information that must be maintained for legal and audit considerations.

Rewritable

Characteristics

Rewritable (also referred to as "Erasable") optical disks have been introduced as a viable mass storage technology relatively recently. More flexible than WORM disks, information stored on a Rewritable Optical Disk cartridge can be erased and re-written. This increased flexibility makes them a more attractive option for many users.

The typical erasable optical disk is 5.25 inches in diameter and encased in plastic, like a WORM disk. Capacities are similar to those of a WORM disk—600 to 800 MBytes (300–400 MBytes per side).

Three separate technologies are currently associated with rewritable optical disks: magneto-optical, dye-polymer, and phase-change. Magneto-optical is the only technology that has reached the production stage, as various problems with the other technologies will require further research before they can be made into marketable products.

Implementation: Magneto-Optical

Magneto-optical technology, as the name implies, uses a combination of lasers and magnetic field effects to store and retrieve data. The disk is composed of a magnetic material, highly stable at room temperature, encased in a plastic cartridge. The value of a bit depends upon whether its magnetic orientation is "north-pole-up" (representing a value of 1) or "north-pole-down" (representing a 0). This is illustrated in Figure 2.

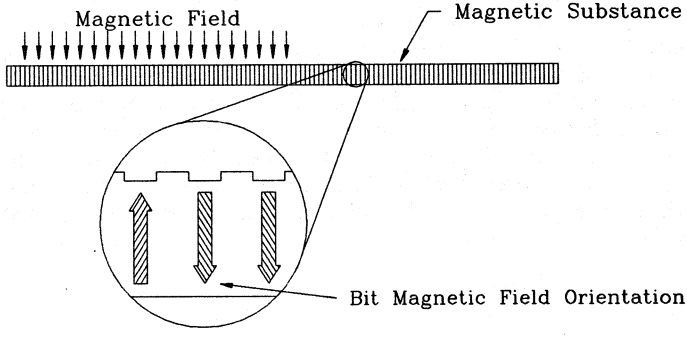


Figure 2: Magneto-Optical Recording

A blank Magneto-Optical Disk (MOD) cartridge has all of its bits pointing north-pole-down. A magnetic coil in the drive produces a magnetic field that points north-pole-up. The strength of a magnetic field required to change the orientation of a bit varies with temperature: at room temperature, the magnetic coil is too weak to induce such a change (it is estimated that, at room temperature, a two ton magnet would be required to change the data on an MOD cartridge). However, at temperatures above 150 degrees Celsius (300 degrees Fahrenheit), the force required to change the magnetic orientation of a bit falls to almost zero, so bits are easily "flipped" by the magnetic coil. To write to the disk, a laser heats a spot on the disk to above 150 degrees Celsius, at which point the magnetic flux can easily be changed by a magnetic head. After the disk cools – only microseconds later – the magnetic flux once again becomes nearly impervious to magnetic fields.

The properties of the Kerr effect are used to read data stored on an MOD cartridge. The Kerr effect states that light will rotate in a particular direction if influenced by a magnetic field. An MOD drive uses this effect by directing a low-power laser at the surface of the disk. The light reflected from the surface will rotate in a clockwise or counterclockwise direction, depending upon the orientation of the magnetic flux of the surface. The read head detects the rotation direction, and sends a corresponding value of "0" or"1" to the computer.

One disadvantage of MOD cartridges stems from the fact that MOD systems must write zeros to the surface before data may be written to that spot. This means that the disk must rotate twice to complete a write operation: once to write zeros, and once to write the desired information. This quirk effectively increases the write-access time of an MOD drive by 40% over read access times.

Implementation: Phase-Change and Dye-Polymer

The phase-change and dye-polymer technologies have similar advantages and disadvantages, though the technologies are slightly different.

A Dye-Polymer optical disk has two layers, each of which is treated with a different organic dye. Each dye absorbs light in a different range of frequencies, so either the top or the bottom layer can be selectively heated by using lasers of different frequencies. To write to the disk, the bottom layer of the disk is heated: it thermally expands upward, deforming the top layer and forming a stable bump. To erase information, the top layer is heated using a laser of a different light frequency which, in effect, "melts" that bump flat again. Information stored on a dye-polymer disk is read using a lower power laser to reflect light off the surface of the disk. The light reflected off a bump (interpreted as a binary "1") is different from that reflected off a flat surface (a binary "0").

A phase-change optical disk has a metal layer sandwiched between two layers of plastic. A high-powered laser is used to heat a spot on the metal layer, changing its molecular structure to a crystalline state (representing a "1"), and back again (representing a "0"). During the read cycle, the laser is directed at the spot using a lower power setting. An amorphous (non-crystalline) spot reflects light with a different intensity than light reflected from a crystalline spot. The bit-state is determined by a photocell, which detects differences in the reflected light.

The dye-polymer and phase-change technologies have encountered a few problems. First, a specific location on the disk can be written only a limited number of times — typically less than 10,000 — which is inadequate for most rewritable applications. Second, the drives use two different lasers (one for writing and one for erasing), making a drive using dye-polymer or phase-change disks more expensive than one using MOD cartridges. However, dye-polymer and phase-change disks do share one prospective advantage over MOD cartridges. Since WORM disks are read using differences in reflected light (as are phase-change and dye-polymer disks), a drive that can read both erasable optical and WORM disks is a distinct possibility. Furthermore, the media are generally less expensive and less susceptible to the environment than MOD cartridges.

Applications

Rewritable optical disk systems have roughly the same capabilities as WORM disks; capacities, access times, and media life are comparable. Therefore, they can be used just as effectively in most of the same applications as WORM drives, except for audit-trail (which requires the unalterable nature of WORM media). The rewritable nature of these drives, however, makes them much more flexible. Since rewritable-optical disk drives "look" like a Winchester hard disk, files can be stored and accessed without using a special archival format. Access to these files is extremely easy; the disk need only be inserted into the drive and mounted.

With Rewritable technology, disks containing obsolete backups and archives can be reused, standard system software can be used with read/write access, and file changes can be stored much more efficiently. Since the disks are rewritable, they may be used in other applications for which a WORM disk would be unsuitable. For such applications, the rewritable-optical system is a replacement or supplement to on-line storage such as a Winchester hard disk or flexible disk drive.

Autochangers

Autochangers, often referred to as "jukeboxes", offer users automated access to a vast amount of information. Autochanger systems typically contain one or two optical disk drives, a storage area for a number of optical disks, and a mechanical arm used to select and load the disks that are stored in the jukebox.

Most jukebox systems use at least two optical drives, to increase the speed at which cartridges can be changed. In a system containing only one drive, a new cartridge cannot be selected and accessed until after the current cartridge is removed from the drive and stored in its proper location. In a system with two drives, a new cartridge can be loaded into drive "B" and access can begin immediately. While the new cartridge is in use, the old cartridge can be removed from drive "A" and stored in its proper location.

Autochanger systems usually have 5%–10% of their capacity on-line at any one time, and have a maximum capacity approaching one hundred GBytes.

Helical Scan Recording

Helical-scan tape stores data using technology that was originally developed by the video recorder and digital audio tape industry. The name is derived from the method by which the tape travels over the head. Traditional tape technologies use a fixed head, with the tape passing (relatively slowly) over the head. Helical scan, however, uses a head that is mounted on a rapidly spinning drum aligned diagonally to the track. As the tape passes over the drum, the head writes tracks of data in a diagonal pattern corresponding to the pitch of the head. This method produces track densities on the order of 1,000–2,000 tracks per inch.

In contrast to traditional computer tape devices, which record longitudinally (along the length of the tape), helical scan tape drives record in diagonal tracks across the tape. This diagonal recording allows data to be recorded more densely and read with greater accuracy. It also results in gentler tape handling, because the higher recording density permits lower tape speeds for the same data transfer rates.

Although helical-scan drives used in the computer industry are very similar to a VCR (Video Cassette Recorder) or a DAT (Digital Audio Tape) player, they require a much higher reliability. VCRs and DAT players typically have an error rate of 1 in 10^6. When an error occurs during a VCR recording, a small extraneous spot may appear on the screen. Similarly, in the case of DATs, a timeout may occur for less than a millisecond. These types of errors are virtually undetectable to human senses. An error rate of 1 in 10^6, however, is unacceptable for mass storage applications. Because of this, error checking and redundancy must be implemented to produce an error rate more in the neighborhood of 1 in 10^{13}.

Helical Scan Advantages

Helical scan tape technology has a number of advantages over traditional tape media. Tapes are very compact (smaller than a deck of playing cards), and offer extremely high capacities—1.3 GBytes for 4mm tapes, and 2.5–5 GBytes for 8mm tapes. The media itself is relatively inexpensive, resulting in a cost per MByte of storage well below one cent.

Helical scan tapes also have a longer archival life (10 years, according to the United States National Bureau of Standards) than magnetic tape, which typically needs to be replaced at least every 3 years. This increases data integrity, saves money by requiring less frequent replacement of media, and saves time and personnel in the maintenance of archived information.

8mm video tape

Characteristics

8mm helical-scan tape systems were derived from commercial Camcorder technology. With a maximum storage capacity of 2.5 GBytes per tape (and 5 GByte tapes looming on the horizon), 8mm media has the single highest storage capacity-to-volume ratio of any mass storage device currently in use (326 MB/in^3). Access time, as with all tape systems, is relatively slow: in the tens of seconds. Burst transfer rates are on the order of 10 MBytes per minute.

The media used for 8mm tape in the mass storage industries is the same lightweight, plastic cartridge used in the entertainment field. Any high quality metal tape from a Camcorder can be used in an 8mm helical-scan tape drive (although data grade tapes are recommended). The 6" by 4" by $\frac{1}{2}$" cartridge fits easily into a shirt pocket. The volume of sales for the entertainment industry has drastically lowered the price of these cartridges to under $10.00 each. This factor, plus the huge storage capacity of these tapes, has driven the cost of storage to less than one penny per MByte.

Implementation

On 8mm tape drives, three heads (two read/write heads, and a servo head to position the tape) are mounted on the rotating drum, which is tilted at a 5° angle from vertical. The drive also has a stationary erase head positioned 1.56 inches from the read/write position in the tape path. In each rotation, one track is written, and the second head read/write head provides a read-after-write check.

To eliminate crosstalk between tracks, fully erased guard bands are left between tracks: this protects against reading data from adjacent tracks, and reduces the need for exacting tape positioning. During write operations, the erase head is always active, "erasing" the portion of tape that is about to be written. This ensures that there are no residual servo or data signals from previous writes. This is shown in Figure 3.

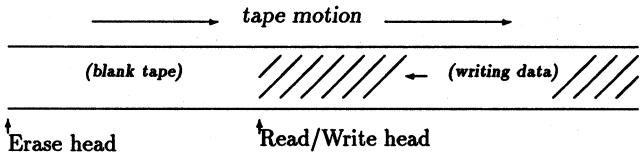


Figure 3: 8mm Tape Writing Procedure

4mm DAT

Characteristics

DATs, or Digital Audio Tapes, are just now being marketed in the computer industry. They are very similar to 8mm tapes, the main difference being that the medium is 4mm wide instead of 8mm (resulting in a maximum capacity that is half that of 8mm tapes). In addition to being half as thick, DATs are also smaller than 8mm tapes (about 3" by 2").

Implementation

The audio version of DAT has two combination read/write heads, where DAT drives for computer applications use two read heads and two write heads (for a total of four heads). DAT drives with only two heads do not have a read-after-write capability, so data reliability is reduced.

On four-headed DAT drives, two diametrically opposite write heads are mounted on the rotating drum, which is tilted at a 6° angle from vertical. In each rotation, each head writes one track at a different angle from the other head, resulting in two tracks written per rotation. Since the heads are wider than the tracks, the tracks overlap each other slightly, with no guard bands between them.

To eliminate crosstalk, each write head is set at a slightly different angle—±20° from the *azimuth* (the axis around which the drum rotates)—so data in adjacent tracks is written at different angles. Each read head reads only the tracks which were written at the same angle as itself, and uses the weaker signals from surrounding tracks to center itself. These read heads are mounted on the drum at 90° angles from the write heads, and provide a read-after-write check.

Because DAT drives were originally developed for audio recording, a special format designed to "overlay" the basic audio format is implemented to allow DAT drives to be used for computer applications. There are currently two competing formats—DDS (Digital Data Storage) and Data/DAT.

The DDS format, originally developed by Hewlett-Packard and Sony, is a "traditional" tape format: files are stored sequentially on the tape, and stored files cannot be modified or updated. The Data/DAT format supports both sequential and random access modes. However, the tape drives are by nature sequential devices—using Data/DAT formatted drives in random-access mode requires excessive tape pre-formatting (which takes about 2 hours). This pre-formatting not only takes time, but wastes a significant amount of tape capacity.

Autofeeders

As with optical disk drives, systems that handle multiple tapes are also available. These autofeeders differ somewhat from optical disk autochangers, because of the way in which they are typically used.

An optical disk autochanger is generally used for random access to on-line storage. Since speed is important in this applications, optical disk autochangers often contain two drives. Helical scan tape drives, however, are not used for on-line storage, and do not offer random access to files.

Since their main purpose is for backup, autofeeders were designed to facilitate the automated backup of systems that are too large to fit on a single tape. These systems typically contain a single drive, and a mechanical arm which is used to sequentially load and unload tapes. The operative word here sequential: tapes cannot be randomly loaded and unloaded; rather, they are loaded into the drive in the order in which they were placed in the autofeeder storage area. As one tape is filled during backup operations, the autofeeder will automatically load the next tape and continue the backup operation.

Helical Scan Applications

High capacity helical scan tape drives are becoming the solution for backup and archival operations. Their extremely high capacities make it possible to perform unattended overnight backup of systems too large for other mediums. With Autofeeders, systems up to 25 GBytes can be backed up without operator intervention.

Helical scan tapes offer another advantage for archival—a shelf life of 10 years, compared to 3 years for other types of magnetic tape media. Due to the method in which information is stored on helical scan tapes, information is not as susceptible to bit migration and print-through. No other medium offers as high a capacity at as low a cost as helical scan tape.

Summary Comparison of Helical Scan Technologies

	8mm 2GB	8mm 5GB	4mm DAT
Recording Density (KB per inch)	45	45	61
Track Density (Trakcs per inch)	819	1638	1869
Areal Density (MB per inch^2)	35	74	114
Capacity (GBytes)	2.5	5.0	1.3
Read/Write Speed (KB per second)	246	500	183
File Search Speed (MB per second)	2.4	37.5	36.0
ECC Overhead	28%	28%	29%
Correction Block (Bytes)	1024	1024	5756

Conclusion

As optical and helical scan systems are refined, their cost, storage capacities, reliability, and speed will all improve. 4mm and 8mm tape manufacturers are already developing helical scan tape systems with capacities twice what is currently available. In the optical arena, products that incorporate a combination of optical technologies are looming on the horizon: units that contain both a WORM and a rewritable optical drive have already been announced, and should be available in the near future. Whatever the future holds, users can be sure that these technologies will continue to offer new solutions to old problems, and gain widespread acceptance as their advantages become more apparent.

REFERENCES:

Freese, Robert P.
"*Optical Disks Become Erasable*"
IEEE Spectrum, February 1988.

Levine, Ron
"*Optical Storage Comes of Age*"
DEC Professional, November 1988.

Parker, Chas
"*Digital Audio Tape*"
HP WORLD, April 1989.

Small, Steven F.
"*8mm and DAT Formats Compete for Tape Technology of Choice*"
Computer Technology Review, Spring 1990.

Vermeulen, Bert
"*Helical Scan and DAT—a Revolution in Computer Tape Technology*"
Hewlett-Packard Company.

PAPER # 5161

MANAGING EMPLOYEE LEARNING

By David G. Robins
Hewlett-Packard Company
Customer Education Engineering
100 Mayfield Avenue, MS/36LC
Mountain View, CA 94043
415/691-5604

Managers are responsible for the productivity of their units. Direct supervisors are concerned with the job performance of each employee. If there is a need for new or improved performance on the job, the supervisor may look to training as a solution.

An executive vice-president of the American Society for Training and Development (ASTD) noted that learning on the job provided fifty-five percent of the U.S. productivity growth between 1935 and 1985, compared to twenty six percent from learning in schools and nineteen percent from the introduction of new capital equipment. The investment in training provided twice the return on productivity as new plant and equipment investments.

This paper deals with a simple concept that will assist in ensuring the high productivity improvement from training. *If the level of learning is high, there is a better chance that there will be some payoff in productivity increase. Direct supervisors must manage employee learning to benefit from high-quality training.*

Supervisors expect that high quality training alone will result in a high level of new or improved performance on the job. There seems to be an imagined (or expected) direct line relationship from training to job performance. Let's look at this relationship and the forces that impact it.

Job Performance

Without getting into a discussion of Maslow's Hierarchy of Needs, let me say that most employees will work on the tasks of a job in some order of importance. That importance is set by situations surrounding the job. The priorities of job performance are set by the employee's perception of the importance of each task and how it will help him or her succeed (see Figure 1).

Job performance, that is the performing of job tasks, is accomplished by the employee according to the priorities set, or perceived to be set, on those tasks.

For example, if the policy says "all output must be inspected according to Schedule X and be 100% free of errors before shipment," but the employees are measured on volume of shipments and not held accountable for errors in shipped materials,

the priorities are reset. This may not be on paper or in formal policy, but in the actual accomplishment of the task there is a new priority.

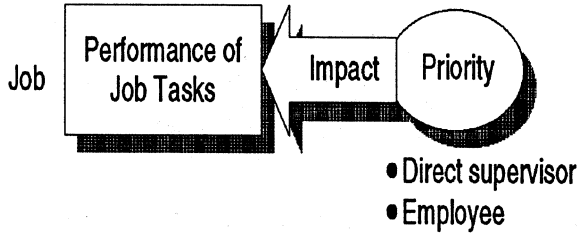


Figure 1

Most task prioritization is set, formally or informally (consciously or unconsciously), by the direct supervisor of the employee.

Priorities are set according to the employee's needs, the interdependence of that task to other tasks and jobs, and the *responses of the immediate boss of the employee.* Positive responses usually move a task up the priority scale.

Training

Training is a job in itself. For training to be effective we usually look to the "quality" of the training. In general terms, training quality depends on the design of the training course, the materials of the course and the delivery of the training to the student. Delivery most often refers to the quality (skills/experience) of the instructor. It also refers to the application of self-paced training media (including CBT, workbook, video, and CD-ROM).

Have you ever been to a training course that was excellent in all the aspects we've just mentioned? Great instructor, excellent materials, perfect design. Let's assume the highest quality of training in all aspects.

The next question is, did the training work? That is, did the students learn not just what was taught, but did the class teach what the students *needed* to improve their job performance? And if it did, did the students apply what they were taught? Did the students even know what they were supposed to get out of the classs in the first place?

My point is that there is not a direct relationship between the quality of training and the job performance. High quality training exists that may have no relationship to the job performance.

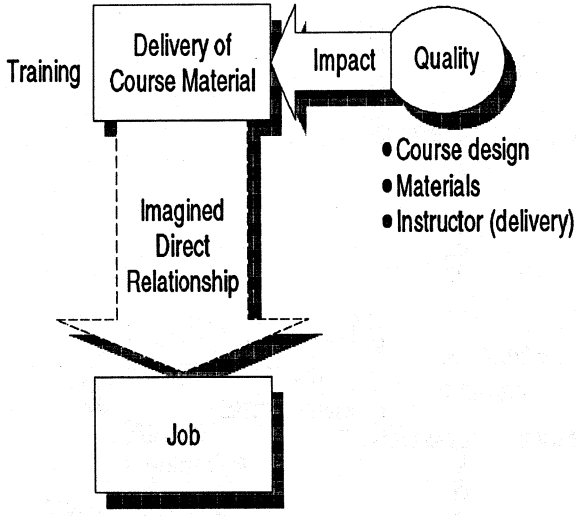


Figure 2

Can training be high quality and the students not learn? Yes, if the perception of the student is that the training is not needed or will negatively affect the job, or if the student is unable to assimilate and use the content.

I have had students attend training classes having no idea why they were there other than "my boss signed me up." I've also had students that assured me that their boss would not let them use anything they were learning in the class because it was not their normal procedure. I must guess that the supervisor either sent them to the wrong class or, worse, the supervisor intends to change the procedure but did not tell the employee prior to the training. Learning is minimized instead of maximized.

Learning

Between Training and Job Performance is Learning (see Figure 3). If learning is well-managed, there is a greater assurance of a payoff from training. Learning is effected by the employee's perception of the importance of the training. If the employees know they will need the content of the training for their job, they are more likely to learn. As a matter of fact, their drive to learn may help overcome what may be lacking in less-than-high-quality-training.

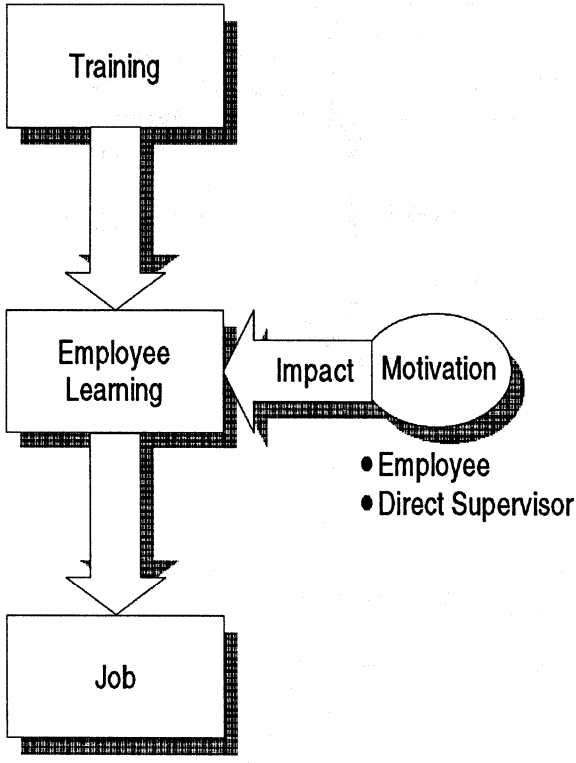


Figure 3

When students are motivated to learn, they become active in the training process. They ask questions. They apply the concepts of the training to their needs and hold the instructor responsible for the content they need. You see this in colleges where students learn in spite of bad teachers, if they need the learning for their major or some other reason.

If the training is high quality and the student is motivated, there is a positive experience for the student, a sense of accomplishment, and an excitement about using the new knowledge or skills back on the job.

The goal of managing employee learning is to improve the motivation to learn.

How to manage learning

CHOOSE THE RIGHT TRAINING: Training is marketed like any other product. Do not just buy the title (or label). Know what your employees need. Do a mini-needs analysis at the job and task levels. Then ask the training organization how they can meet those needs. This applies to internal and external training.

If knowledge is needed, a seminar may be the correct format. But if a skill is needed, the student should be involved in some lab exercises to build the skill. Are labs offered?

Is the training generic? Does it transfer to your employees' needs or must it be customized? Should a trainer be brought onsite to develop skills on your actual equipment and applications? Should the employees be sent offsite to leverage the experience of students from other companies?

Is the training high quality? Is the quality of the training consistent so your future employees are assured of the same learning?

COMMUNICATE WITH YOUR EMPLOYEES ABOUT THE TRAINING: Do they agree with the need for training? Do they know why they have been selected to be trained? What will they do with the training? What are your expectations? How will they know if the training is good, before it is completed? How will they report back to you? When will they put the training to use? What benefits will the *employee* get from the training?

It is important to meet with your employee BEFORE the training so the motivation level is high. Create excitement about the upcoming experience. Make it important to the individual in a positive way.

If the training takes more than one day, arrange a phone appointment with your employee to check on the training. DO NOT discuss work other than the training.

Cover the employee's workload and let them know it is covered. I've seen many students phase out of the learning on the third day of a class as they realize their in-tray is probably overflowing and "Monday will be a day of catch-up and overtime."

FOLLOW THROUGH AFTER THE TRAINING: Has time been set aside for the employee to use the training right away? What is learned and not used is soon forgotten. Have you allowed for a learning curve? Do both you and the employee realize that productivity (speed) may be down as the new skills are being honed? If not, employees throw out the new in favor of the tried and true and comfortable.

Is the new skill measured as part of a performance evaluation? Has the job description been updated to include the new tasks? Will the employee

realize any benefit from the application of the training/learning and does he or she know what that is?

Conclusion

The direct supervisor of the employee plays a primary role in causing an actual return on the training investment. Although upper management may espouse the importance of training, and authorize the expenditures to pay for the training, it is the direct supervisor that sets the level of importance to tasks within a job. It is the direct supervisor, therefore, that motivates or de-motivates the employee in a training situation.

High quality training is important. It must also be the "right" training. But a de-motivated student will not learn no matter how good the course and materials and how hard the instructor works. The employee must be motivated to learn, and motivated to use the learning on the job. The direct supervisor can manage that learning.

TITLE: Maximizing Your Training Dollars

AUTHOR: Chosen Cheng

Hewlett-Packard Co.

5301 Stevens Creek Blvd.

Santa Clara, CA 95052

(408) 553-3946

FINAL PAPER NOT AVAILABLE, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO. 5162

Paper # 5163
CHANGE: Wish It Were Easy?

Hewlett-Packard Company
Human Resources
19447 Pruneridge Avenue, MS 48SC
Cupertino, CA 95014

We are in a world which has become capable of creating and introducing more changes (big and small) into our lives than once imagined possible. You'd think with all this practice, change would have become easy. When we don't feel impacted by a given change it is far easier. However, as is so often the case, when there are direct or even indirect consequences to us personally we experience resistance.

As human behavior would have it, fear (an instinctive survival mechanism) still serves us in the face of modern day changes. Even "advancement" (say in the form of technological innovation) can trigger fear based resistance. There are several key sources of fear which can be expressed in the form of resistance. Fear that we will not be capable of making the change(s) successfully. Fear that the change(s) will cause us to lose something familiar and valued. Fear that the promises of the given change(s) will not manifest. And the fear we experience in the face of excessive uncertainty, confusion, and ambiguity.

Effective change management means anticipating and working with natural resistance and the fears that fuel it. Knowing that resistance is a core human response to the threat(s) posed by change only helps us when we recognize that every individual defines his or her own "threat(s)" uniquely. "Frame of reference" is the unique way that we each perceive our situation based largely upon our past experiences. This will shape how threat and opportunity are defined and responded to.

As leaders of change we must begin by understanding not only the potential direct and indirect impacts of our proposed change(s) on others but also the frame or frames of reference through which these impacts will be seen and felt. This is no easy task and there is no special formula. What instead we will offer is a three-part model of change which acknowledges that resistance is a natural part of the change process that can (and needs to be) anticipated, understood, and managed.

CHANGE MODEL

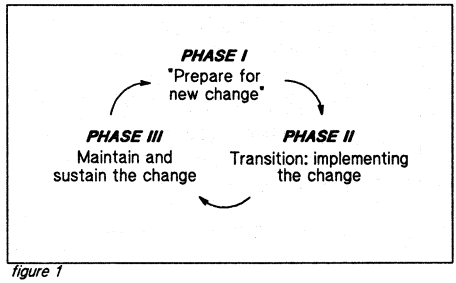


figure 1

The first phase of our change model involves preparing for change. During this upfront period, your work is to think through the intended change in order to clarify the reasons that are driving the change. As you may have experienced, providing a sound and logical rationale alone is not enough. Once you have focused upon the impetus

for change, it is time to assess how others, through their frame of references, may experience your proposed change. Begin by asking yourself: who are the people who may directly or indirectly be impacted by this change?; who has the ability to block this?; and who must actively support this? What might they have to lose or let go of in order to support this change? How might they perceive this loss? Are they experiencing enough pain or dissatisfaction currently that will reinforce their willingness to let go of a part of their status quo?

Loss is perhaps the most important aspect to clarify and anticipate during the preparing for change phase. This is because resistance will be ignited at the fear of loss. When the loss(es) are understood, acknowledged, and allowed to be worked through they can be let go of. However, when losses are denied, invalidated, or simply not recognized as part of someone's experience of the change then resistance can block, even paralyze a proposed change effort.

The preparing for change phase also includes determining whether in the eyes of those experiencing the change, there is enough future gain or payoff seen to justify the effort and loss which might be required to both make and sustain the change. Involvement on the part of those who must implement the change during the preparation phase is well worth the investment. Their involvement allows you the opportunity to share the intent of the change as well as the rationale. But most importantly, you have the chance to learn from the questions and concerns that surface what it will take to successfully implement this proposed change. When you recognize that these people are not the obstacles to your change effort but rather the enablers, time spent upfront in preparation can clearly be justified.

The second phase of our change model involves leading through the change. Often after a change is announced, the management of the transition from "what used to be" to "what in not yet" is left to chance. This time of transition is far more challenging and tenuous than is often recognized. To the extend possible during phase I, the design of your change with "user" involvement has anticipated and tried to obviate critical transition issues. Even when this is done well, resistance will be a natural and expected part of "transition". Individuals' capacity to accept and adapt to change lags behind the time it takes to make the formal change itself. Hence phase II really concerns helping people process and adopt change both in their thoughts and their actions. Figure 2 describes the important characteristics of this transition period.

CHARACTERISTICS OF THE TRANSITION PHASE

1. Low stability
2. High emotional stress
3. High, often undirected energy
4. Control as a major issue
5. Past patterns of behavior become highly valued
6. Conflict increases
7. Fear/avoidance

figure 2

Given the characteristics of the transition phase, largely characterized by uncertainty, confusion, and ambiguity, it is critical to create mechanisms to understand what barriers people are experiencing within or outside themselves as they move into implementing the change. Paying attention to resistance through effective two-way communication forums will allow the uncertainties, fears, and concerns that can impede and derail effective change to be surfaced and worked with.

The third phase of our change model acknowledges that feedback and fine tuning are important in order to maintain and sustain the change. Certainly strong two-way communication and involvement on the part of those with a vested interest in the change have been emphasized up to this point. We recommend at this phase that you assess how effective the change process and results have been relative to the intended aims. With the understandings derived through soliciting feedback and doing an overall asessment, you are well positioned to make and necessary course corrections.

When you reflect on the change efforts that you are leading or have lead in the past, are there some things that you pay too little or too much attention to? We suggest that top on your list of "success factors" are: 1) viewing and managing change as a process rather than an event; 2) attending to upfront planning, preparation, and "stakeholder" involvement; and 3) expecting, acknowledging, and working with resistance.

There are countless factors that influence the success of a change effort. However, the "human side" of change is most often the key differentiator between a change effort that succeeds and one that does not. Our premise is simple, effective change management requires both understanding change as a process and understanding how and why people respond to change. Your participation can go a long way to determine whether people hinder or enable the change.

5164

INFORMATION MANAGEMENT TECHNOLOGIES INTO THE 1990'S

Orland J. Larson
Hewlett-Packard Company
19091 Pruneridge Avenue
Cupertino, California 95014
(408)447-1197

ABSTRACT

Enterprises have come to depend more and more on the accessibility, accuracy and timeliness of information. During the next few years, new database technologies will provide solutions to many of the problems and difficulties facing today's MIS and user departments.

This paper begins by reviewing the changing world of information management and the challenges facing MIS. This will be followed by the major trends associated with information management over the next several years. These include the continued acceptance and importance of relational database technology, the increased interest in distributed database applications and the emergence of the new object- oriented database management capabilities. Finally, this paper will address the significance of these technologies in the cooperative computing environment.

INTRODUCTION

THE CHANGING ROLE OF DATA PROCESSING

Before looking forward into the 1990's, we should first look back at how data processing has evolved over the past twenty years.

The general trend of the 1970's was the use of centralized computers and resulted in systems that were often difficult to use, inflexible and usually did not meet the end user's needs. Database management systems (mostly hierarchical and network) became widely used and provided the basis for on-line, interactive applications. In addition, the computers and operating systems provided programmers the capability of developing applications on-line, while sitting at a terminal

and interactively developing, compiling and testing these applications. The end users were also provided easy-to-use, on-line inquiry facilities to allow them to access and report on data residing in their databases.

During the 1980's, the emphasis was on the decentralization of data processing. This includes the proliferation of personal computers which has resulted in both the "islands of automation" and the corresponding "islands of information" problems. This in turn resulted in reduced control of corporate data for the MIS department. In addition, relational databases became commercially viable and experienced wide acceptance even though performance was often an issue. Relational database performance has now improved significantly; and they are currently proving effective in on-line transaction processing (OLTP) environments. Software tools such as 4th generation languages (4GL's) continue to be used successfully as an effective way of developing applications through the concept of information systems prototyping. This required that the end user be more involved in the development of systems and has resulted in more effective systems that meet the users' needs. This has helped to reduced the backlog of applications but usually also has contributed to the "islands of automation" problem.

As we move into the 1990's, relational database will continue to gain wider acceptance. It is the enabling technology and the basis for distributed database management, which provides transparent access to data which is distributed over several sites.

There are also new technologies called object oriented database (OODB) and object oriented programming systems (OOPS), which will manage more complex data structures and will result in improved programmer productivity and more flexible systems.

An additional technology, cooperative processing, is evolving which will help integrate those "islands of automation" back together and allow for the data and programs to be accessed and shared in a cooperative computing environment.

THE IMPORTANCE OF SQL

According to a recent Gartner Group Report, in 1988 only about 7% of the applications developed used relational database or Structured Query Language (SQL). However by 1992 their prediction is that 65% of applications developed will use SQL.

There is no doubt that SQL will be the basis for applications developed in the 1990's. One of the main advantages of SQL is data independence or the immunity of applications to changes in storage structure and access strategy. Another main advantage of SQL is the simplicity of the underlying relational model which is the easiest to understand - at least at the most basic level. In this model, data are represented as tables, with each horizontal row representing a logical record and each vertical column representing one of the attributes, or fields, of the record.

The following are the key points associated with relational technology:

* Relational concepts are easy to understand and use.
* SQL is a multifunctional language
  - Database definition and creation
  - Data retrieval and manipulation
  - Authorization and security
  - Transaction management and recovery
  - Database environment management and restructuring
  - Interactive and programmatic use
* SQL allows you to specify which information you want - not how to retrieve it.
* SQL increases programmer productivity and raises programming closer to the level of problem solving.
* Data independence is ensured and minimizes maintenance of programs
* Data access can be automatically optimized as the database structure changes.
* The DBA has unprecedented power and control over the database.
* New systems can be implemented much faster.
* SQL assists in cross-system connectivity.
* Relational databases provide a cost effective, powerful solution.
* Basis for a true distributed database environment.

There are, however, some areas of SQL that need improvement. The current SQL standard is missing some important features and some of the standard features are implementor defined. In other words, no vendor fully supports the complete "standard" and no two SQL implementations are exactly alike. These inconsistencies will lessen as new levels of standards evolve. Organizations like the SQL Access Group are also working hard to resolve these problems.

As we enter the 1990's, relational database is becoming the dominant technology in today's information management marketplace. There are several enhancements planned to improve functionality and performance. It eventually will be appropriate for most applications and gain wide acceptance by all users.

Relational databases can improve the quality, control and accessibility to your organization's extremely important and valuable information resources. It can result in an improved competitive position by aiding business analysis that can help to determine ways to improve products and services.

Unlike non-relational database environments, relational databases adapt easily to dynamic business requirements. In addition, unrestricted access to important data means better information for more effective decision making.

Relational database can also have a positive effect on many MIS development environments by reducing the application backlog and reducing the time and cost required to develop applications. The improved database flexibility and ease of change can also result in a significant reduction in the maintenance of applications.

Overall, the use of relational technology can increase the MIS professional's effectiveness and productivity, which results in improved user satisfaction and confidence. Choosing relational now will position your organization to take full advantage of the technological advances of the 1990'S.

DISTRIBUTED DATABASE

One of the hottest topics in the commercial database world is the growing trend towards the use of distributed database management systems. After many years of research, distributed databases are becoming more viable. However, there is still much to be done to provide more than just read access to distributed data. Chris Date, one of the world's leading experts on relational database, recently presented a paper (see reference 3) in which he provided a working definition of distributed database. "A distributed database system is a system involving multiple sites connected together in a communication network, in which each site is a database system in its own right, plus a user at any site can access any data in the network exactly as if the data were all

stored at the user's own site. Thus, a DDB is a virtual DB whose components are physically stored in a number of distinct real databases at a number of distinct sites."

Chris Date follows this working definition with an "alternate" or more elaborate definition. "A distributed database system is a system that allows an arbitrary collection of relations, from an arbitrary collection of databases, on a variety of different machines, running a variety of different operating systems, connected by a variety of different communication networks to function as if they were all stored in a single database on a single machine. The user is completely insulated from all details of distribution."

Distributed databases can allow the structure of the database to mirror the structure of the company, while simultaneously solving the "islands of information" problem. Some additional advantages include local control of local data, accessibility to remote data, increased capacity, incremental growth, data availability, efficiency of storage, flexibility and cost effectiveness.

There are also some potential problems or disadvantages associated with distributed database such as the complexity of implementation - but this is the vendor's problem. Some additional potential problems include the problem of how to design systems for distributed environments, the complexity of administration and control, the impact on local operations, the political problems dealing with the ownership and protection of the data and the possibility of a node or line "crash". In addition, solutions that are appropriate in a centralized environment may frequently not be appropriate with distributed systems.

Chris Date is the author of 12 rules of a distributed database system. He begins with the fundamental principle or "rule zero" that states, "To the user, a distributed database system should look exactly like a nondistributed system". The subsidiary rules follow:

1. Local autonomy
2. No reliance on a central site
3. Continuous operation
4. Location independence
5. Fragmentation independence
6. Replication independence
7. Distributed query processing

8. Distributed transaction management
9. Hardware independence
10. Operating system independence
11. Network independence
12. DBMS independence

These rules are fairly self explanatory and will not be expanded in this paper. No current DBMS vendor adheres to all of these rules. A few vendors claim adherence to rules 1-8 and almost none to rules 9-12.

Hewlett-Packard is using a phased approach for developing a distributed database management system that will be rolled out in the early 1990's. The following is a summary of that approach:

* REMOTE DATABASE ACCESS (ALLBASE/NET) - Program can read/update a remote DB without coding for communication and remote processes.
* FOREIGN DATABASE ACCESS (e.g. IBM's DB2) - Program can access multiple vendors databases without coding for DBMS differences.
* DISTRIBUTED UPDATES WITH TRANSACTION MANAGEMENT - Enhanced transaction management to support updates to multiple DB environments.
* MULTI-REMOTE DATABASE ACCESS - Program can read and update more than one remote database at a time.
* SNAPSHOTS - Enables a user to copy all or part of a table from one database to another, optionally this table could be refreshed.
* DISTRIBUTED JOINS - Enables users to "join" data which resides in separate databases.
* PARALLEL QUERY EXECUTION - Ability to execute multiple queries to multiple databases at the same time.
* REPLICATED DATA - Enhanced availability and performance through multiple copies of data, with automatic synchronization of copies.
* PARTITIONING - Enhanced performance and availability through partitioned tables.

Some of the issues regarding distributed database that will have to be addressed in the 1990's include:

* Distributed query optimization and decomposition

* Fragmentation, recombination and optimizability of data
* 2-phase commit and recovery
* Referential integrity across sites
* Management of replicated/partitioned data
* Controlling authorized user access
* Update synchronization
* Degree of Transparency
* Flexibility to move data around the network
* Cost of mainframe vs. mini vs. micro MIPS
* Provision of foreign (non-HP) DBMS gateways (gateways are a way of processing data in a foreign DBMS or file system)

There are some factors that will help propel the distribution of data. These include company mergers, the downsizing of computers, the increased database needs and the general industry push toward distributed and cooperative processing. A company with distributed operations will gain competitive advantage through the support of distributed databases.

OBJECT-ORIENTED SYSTEMS

Each decade, one or two key advances emerge to change the practice of software development. Object-oriented systems and methods are rapidly entering the mainstream of software engineering and systems development. Leading consultants are heralding object-oriented approaches as one of the most important trends to affect businesses in the 1990's. But even among its strongest advocates, disputes abound over key issues, content, and definitions of the object-oriented approach. Object-oriented technologies are moving out of the academic world and into the business world.

With the object-oriented approach, processes revolve around the data, not the other way around. Using the traditional approach, programs are structured around data rather than procedures. For example, when using a traditional programming language, parameters are used to pass data structures and values between routines. The object-oriented approach attaches routines to data structures. In other words, the behavior of the data is kept with the data. This is called encapsulation.

An object-oriented programming language allows the programmer to define and manipulate objects. Some object-oriented programming languages are extensions of classical languages - C++ and ObjectiveC are in this category. Others are newer languages, e.g. Smalltalk and Eiffel. In addition, object-oriented extensions are being proposed for ANSI COBOL.

An object-oriented DBMS also supports the definition and manipulation of objects, plus providing the classic DBMS functions of persistent storage, transaction management, concurrency control, security, backup and recovery.

A "message" is an important concept used with the object-oriented approach. It is defined as a request sent to an object to change its state, or to return a result to the sender. Objects respond only to well-defined messages. The only information needed to use an object is knowledge of the messages it can receive. An object-oriented program is a flow of messages among cooperating objects.

Messages ensure the modularity of a system. To interact with any object, you only need to know what messages to respond to, not how the object is represented. Knowledge of how an operation is accomplished is of interest only to the programmer responsible for the definition of the object itself. Messages make an object's functionality available to other objects, while hiding the implementation details.

Maintaining and modifying software has been a real drain on programming resources. Maintenance programmers must understand a complex system well enough to fix its problems or enhance it. However, these programmers often did not develop the code, and often operate without adequate documentation or guidance. Changes often introduce new, unanticipated problems to the system. A programmer working with existing code must read and understand it; this may require a mental translation back into the original design specifications, which is extremely difficult for complex or poorly-coded systems. An additional concern is that programming languages allow unchecked access to data structure internals.

Objects can dramatically improve the problem of maintenance. Modularity and encapsulation limit interdependence, allowing changes that do not disrupt the rest of the system. Objects' natural organization make it easier to learn and understand relationships between parts of a system. The original programs are easier to write and debug and fewer errors occur. Programs read like

designs, making changes clear and easy to make. Reuseability is an important advantage of using objects and libraries of these software components provide leverage. What has been written once need not be written again. Model features like inheritance allow existing components to be incrementally modified to suit changing needs. Together the representational advantage of encapsulation and the features of inheritance dramatically improve software development and can greatly improve programmer productivity.

Libraries of high-quality, tested software components will radically alter the way software is written. Software will routinely consist of a series of software components glued together. Application programming will no longer mean rushing to a text editor to begin coding. It will require understanding the capabilities and restrictions of available components, plus knowing how to combine them.

Some of the challenges facing object systems include the time it takes to learn about existing libraries of software components. Programmers also may resist accepting this new approach. Objects also consume more resources, however emerging 80486 PC's and RISC workstations will help to alleviate this problem. Applications with promising potential include: prototyping, user interfaces, graphics, telecommunications, geographic information systems, computer aided design (CAD), and computer aided manufacturing (CAM).

Object oriented products are still in their infancy and the commercialization of object-oriented technology has barely begun. Now more suitable for advanced technology projects, object database systems should become viable for commercial projects over the next few years and widespread adoption by the mid-1990's.

Hewlett-Packard has developed a prototype of an object-oriented database management system (ODBMS) called Iris. Development started in 1984 and Iris has been presented and demonstrated at several major conferences in the past few years including: SIGMOD (Special Interest Group on the Management of Data) in June 1988 in Chicago, OOPSLA(Object-oriented Programming, Systems, Languages and Applications) in September 1988 in San Diego, at the Patricia Seybold Forum on Object-oriented Technology in April 1989 in Boston and DB/EXPO'91 in March 1991. Iris is in use at universities in the U.S. and Europe.

To be successful, ODBMS technology must integrate well with existing data management systems. Using query interfaces similar to SQL will dramatically reduce retraining costs. Accessing data outside the ODBMS will allow users to use their existing applications in concert with new ones developed using an ODBMS.

Object-oriented environments herald the dawn of new programmings paradigms. Business people will be empowered to perform tasks that, in the past, required professional programmers. Programmers will be empowered to design complex applications in smaller, modular, more fool-proof pieces.

Neither end users nor application programmers will need to concern themselves with the mechanics of networking, peripheral support, or file handling. Object based architectures lend themselves to the creation of a much richer information environment. Digitized voice, music, images, video clips, and animation will begin to populate our information systems.

COOPERATIVE COMPUTING

The environment in which today's business must operate is changing quickly and becoming more complex. To meet the challenges produced by this changing environment, organizations need greater amounts of information to make the key decisions required for success. Keeping pace with the rapid change that is occurring means gathering information and making decisions faster than ever before.

As we look at the computer industry, we are about to embark upon the next revolution in computing. This is not a unique case. There have been multiple revolutions in the computing industry and computation in general, going back to the early days of mechanization, tabulation and so on, through the first computer mainframes. The mainframe was a very centralized processor, still oriented toward batch and was really a carry-over from punchcard tabulation systems. The next move was into mini-computers and distributed processing followed by the personal computer revolution. This caused an explosion of workstations and personal computers on people's desks, which fueled distributed processing and distributed computing. Then came the communications revolution, where the objective was to integrate all this computing power in the corporation in a way that moves information around and enables different types of devices to participate in the

solution of business problems. This has become the basis or launching point for the next revolution of cooperative computing.

Cooperative computing is the notion of tying together all the information computation resources in a corporation into a single entity, and making all those things interact in some efficient manner, transparent to the end-user in such a way that each user has access to all the information computation resources in the network as though they were local to the user's workstation.

NewWave Computing is Hewlett-Packard's implementation of the industry trend towards cooperative computing. It includes HP's vision of the future of computing: a network of heterogeneous computers that can work together to solve a single problem and are extremely easy to use.

To support cooperative computing requires a strategy that addresses the technological implementation needs. The NewWave Computing Architecture brings together systems and servers, easy to use workstations, industry standard networking and perhaps most importantly, the integration of the three through innovative software.

This paper has addressed many of the important technologies that are essential to the cooperative computing environment. ALLBASE/SQL is HP's strategic relational DBMS of the future for cooperative computing. ALLBASE/SQL runs under both MPE and HP-UX. Application development environments for both of these platforms include powerful tools from both HP and Third Party vendors. ALLBASE/SQL will also be the basis for distributed database technology in the future. ALLBASE/NET is the first phase of HP's distributed database technology providing remote data access and uses HP's networking capabilities.

HP ALLBASE/Turbo CONNECT provides coexistence between ALLBASE/SQL and TurboIMAGE by allowing ALLBASE/SQL applications to read TurboIMAGE databases. With ALLBASE/Turbo Connect, customers can preserve and leverage investments in TurboIMAGE/XL applications while reaping the benefits of relational technology through ALLBASE/SQL. A single industry-standard SQL interface allows access to both types of data.

ALLBASE/DB2 CONNECT allows ALLBASE/SQL to access (read and write) DB2 databases on IBM mainframes. Access to DB2 from both PCs and HP3000s will be supported from ISQL

(Interactive SQL) and two PC-based products: Information Access and NewWave Access. ALLBASE/DB2 CONNECT is the flagship product by which we are launching connectivity to non-HP database servers, and extending the PC-HP3000 environment to include IBM mainframes.

The concept of "objects" in the HP NewWave environment is similar to the object-oriented capabilities described in this paper. The Object Management Facility (OMF) is one of the main components of the HP NewWave environment. The OMF tracks all data in the PC, whether it be text, graphics, spreadsheets, scanned images, even voice. These objects are represented as icons which can be combined into compound documents containing different types of data.

The OMF allows users to create "hot-links". Hot-links allow users to share data between different reports, memos, even file folders. When data is changed in one place it is updated automatically in other shared documents throughout the system.

The OMF binds applications and data together to form "objects". By double-clicking on an object, you can simultaneously load the application and call up the desired file, eliminating the need to find file names and their directories.

SUMMARY

The important technologies briefly addressed in this paper: SQL, distributed database management systems and object-oriented systems are extremely important to the future of the cooperative computing environment. HP's NewWave Computing strategy is focused upon helping our customers meet the challenges of today's changing business environment. We believe that these technologies and strategy, coupled with HP's commitment to industry standards and reputation for high quality, reliable systems, can provide our customers with the solutions they will need into the next generations.

REFERENCES

Brown,A.J.,"Distributed Data Management From HP",Proceedings INTEREX HP Users Conference, San Francisco, CA, September 11-14, 1989.

Connors, T. and Lyngbaek, P.,"Providing Uniform Access to Heterogeneous Information Bases",Software Technology Laboratory,Report STL-87-16, Hewlett-Packard Laboratories,December 4, 1987.

Date, C.J.,"Distributed Database Technology", Proceedings DB/EXPO'89, San Francisco, CA, January 31 - February 2, 1989.

Eunice, J.,"Leaders in Object Databases",D.H. Brown Associates, Inc., December 1988, 400 Benedict Avenue, Tarrytown, NY, 10591, (914)631-7859.

Fishman, D.H., Beech, D., Cate, H.P., Chow, E.C., Connors, T., Davis, J.W., Derrett, N., Hoch, C.J., Kent, W., Lyngbaek, P., Mahbod, B., Neimat, M.A., Ryan, T.A., Shan, M.C., "Iris: An Object-Oriented Database Management System", ACM Transactions on Office Information Systems, 5(1), Jan 87.

Larson, O.J., "Strategic Importance of Relational Database Technology", Proceedings INTEREX Computing Management Symposium'89, Nashville, TN, March 8-10, 1989.

Perreault, R., Rotzell, K., "Object Database Management Systems", INTERACT, April 1990, pp.114-126.

Schussel, G, "Database Developments in the 5th Generation", Proceedings 1989 Database and Cooperative Processing Symposium, Washington, DC, March 20-22, 1989.

Stein, J., and Maier, D., "Concepts in Object-oriented Data Management", Database Programming and Design, 1(4), April 1988, pp.58-67.

Tash, J., "Application Architecture and the Strategic Role of Relational DBMS", Database Decisions, Inc., 41 Marcellus Drive, Newton, MA, 02159, (617)332-3101.

White, C., "The Importance of SQL", Proceedings DB/EXPO'89, San Francisco, CA, January 31-February 2, 1989.

White, C., "Emerging Technologies for The 90's", Proceedings DB/Expo'91, San Francisco, CA, March 25-28, 1991.

PAPER NO.

5166

TITLE

I.T. Strategy and Open Systems
How to Succesfully Manage the Open Systems Environment

AUTHOR

Mike Anniss

COMPANY

Hewlett Packard
Amen Corner
Cain Road
Bracknell
Berkshire
England
Tel 0344 360000

A. ABSTRACT

The key to success in the management and implementation of Open Systems solutions is a comprehensive and effective understanding of both, the business needs that the systems are addressing, and the capabilities that an Open Systems based technical solution provides. In order to provide this understanding, a mechanism is required to clearly lay out the issues, and provide for a powerful evaluation, analysis, and description of both, the resultant solutions, and the reasons for those solutions. This mechanism can be provided by a structure called the "I.T. Framework" which details a powerful, logical and complete step by step approach to the transformation of defined business needs to the definition and implementation of system infrastructure solutions.

This paper will:-

. Describe the I.T. Framework and detail the steps to be followed in utilising it.
. Define Open Systems in terms of both "Open Platforms" and "Open Software" and the consequent implications.

B. THE I.T. FRAMEWORK

B.1 Introduction.

At Hewlett Packard we have provided customers with a number of consultancy services to help them understand the best strategic and technical direction for their systems environment. There are a number of strategic planning processes available which look at the higher level strategic questions that need to be answered in order to define that systems environment. The commonly addressed issues are:-

. What business needs should be addressed by the organisation as a whole?
. In what way could business needs be addressed by the use of systems based solutions?

These are addressed by services that have such titles as "Business Planning" and "Information Systems Strategy Planning". The result of these services is normally a report and/or programme that defines the way that systems should be viewed, how application areas should be split up and addressed, and the prioritisation of the steps to be taken to implement the strategy. Hewlett Packard provides these services in a variety of forms, and in delivering this consultancy to customers we have found that

there is a set of issues not fully addressed by these processes. These issues relate to the need to identify:-

- How the transformation from the identified strategic direction to specific products and services is accomplished.

There are three specific sets of sub-issues here:-

- What should the common technical infrastructure look like?
- How should the resources be organised to deliver the required systems environment?
- How can the required function and data for specific areas be delivered?

The second and third questions are normally addressed through, the definition of the organisation's structure, and the normal software development/implementation process. The use of structured and formal methodologies (e.g. SA/SD, IEF, JSD, VDM) provide for a detailed view of the functional and data aspects of business needs and transform these into specific solutions. These methods generally have either no specific way of defining the technical infrastructure, or a poorly defined process that focuses heavily on the specific function or data to be supported. The wider issues of organisational integration and effective evaluation of technical infrastructure possibilities tend to get lost in the rush to develop specific applications, and if not lost, there is rarely a structured logical process that is followed. There is rarely effective documentation produced so that the decisions made can be understood and the situation re-evaluated when the business needs or technical options change. An approach has been developed that addresses this technical infrastructure question and includes a mechanism for facilitating the key issues of how to analyse, derive and document the technical infrastructure required by an organisation. The full analysis is based on a process called I.T. Strategy Planning and the mechanism that addresses the technical infrastructure issues is called the "I.T. Framework". This framework has been found to be a very simple, but powerful method of assessing technical infrastructure issues and then describing them in a way that allows for an easy analysis of future changes in both business needs and technical options.

B.2 What does the I.T. Framework Address?

The I.T. Framework provides a structured way of defining both a future and current IT infrastructure. It details:-

- The business requirements to be addressed.
- The technical considerations that may have an impact on meeting those requirements.
- The risk levels associated with products and services.
- The solution types to be evaluated.
- The solutions to be evaluated.
- The products and service solutions adopted.
- The reasons for the choice of those products and services.

The I.T. Framework is used as an analytical and evaluation tool that enables a controlled, documented and effective transformation from the desired information systems needs to the products and services that can best support those needs. Once the I.T. Framework is in place it becomes an evolving document that is changed as either business requirements or technical options develop. In this way the I.T. Framework becomes a dynamic document which follows the needs of the organisation as it changes.

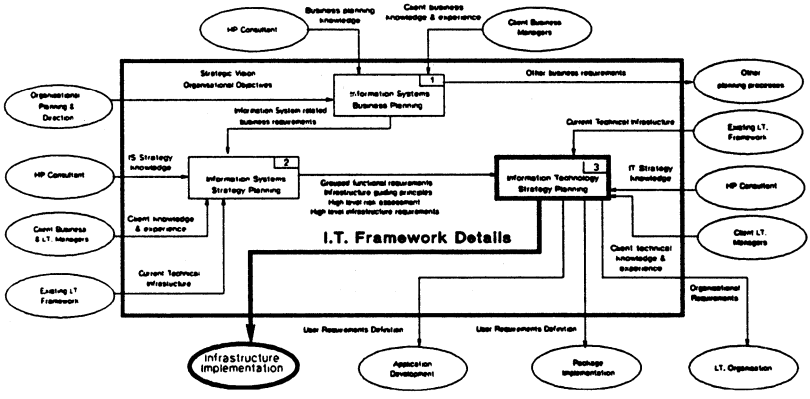


Fig. 1 The I.T. Framework within the overall Strategic Planning process.

The I.T. Framework addresses a number of issues :-

- It provides an overall infrastructure description within which I.T. developments can be coordinated.

- It enables the common needs of the organisation to be taken into account when

parochial developments are taking place, so that developments reflect the needs of the organisation as a whole.

- An integrated policy enables a more efficient use of resources within the organisation. Specific skill sets enable I.T. personnel to move between operations easily rather than having barriers produced by many areas using different methods and products.

- It provides a mapping from the business goals of the organisation via the I.S. Strategy (where it exists) down to the technical implementation needed to achieve those goals. The I.T. Framework can be easily re-evaluated when either the business needs or the technical possibilities change. In this way the organisation can understand :-

  - How technology is used to support the business goals.
  - The risk levels that I.T. solutions should adopt.
  - The technology and the suppliers upon whom the organisation is potentially investing its future.
  - The potential impact of new technology.
  - The technology adjustments that may need to follow changes in business needs.

- The I.T. Framework provides a description of the current state of I.T. in the organisation and provides the information needed to evaluate changes and identify the impact of possible changes. Any changes are reflected in the I.T. Framework which is then used as the new description of the I.T. infrastructure for the organisation.

- The I.T. Framework aids communication between the different groups in an organisation because of its clear logical statement of the derivation of the infrastructure solution that an organisation is adopting. It provides an often missing link in a organisation between the technologists, the business units and the executive decision makers. It enables those at different levels in the organisation to make the link between global strategy and the specific implementation of technology that supports the business.

B.3 Open Systems and the I.T. Framework.

Open Systems provide organisations with a number of benefits:-

- Portability - freedom of choice in hardware vendor selection and potentially in

software selection.

- Interoperability - integration between different applications and databases.
- Scalability - a wide range of performance capabilities.
- Solution Availability - access to a large range of application solutions.
- Fast Technical Advance - the result of many companies trying to better each other generates fast technological change.

Together with these advantages comes a major problem:-

- The vast choice makes the definition of the appropriate solution much more difficult.

As shown in Fig.2 this wide range of choice means that while the chance of an optimum solution being available is greatly enhanced by the wide availability of Open Systems based solutions, the same wide range of choice may make it difficult to identify that optimum solution from the much larger set of possibilities. The I.T. Framework greatly aids this process of refinement towards the solution. Initially the I.T. Framework will enable the derivation of solution possibilities based on the organisation's requirements, to be described in a clear unambiguous way. This enables the high level decision of whether to adopt open or proprietary based solutions to be made in a way that:-

- Directly relates to the business needs of the organisation.
- Highlights the technical issues.
- Identifies the levels of risk that the solutions should follow.

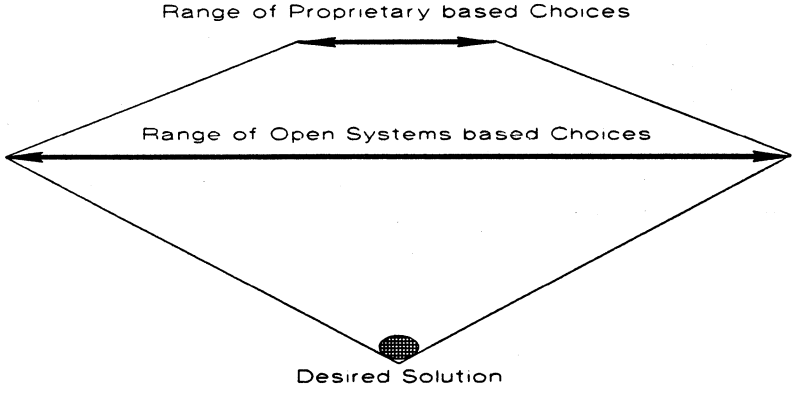


Fig.2 Open Systems Choice

It then makes clear the set of requirements that each solution should be evaluated against and can be used in defining solutions for an organisation wide infrastructure or a specific application area.

The I.T. Framework is thus of particular use in the evaluation and definition of the technical infrastructure needs of an organisation because it enables clear and effective management of the wide range of choices. In addition the framework enables the on-going change in both business need and technology to be clearly and quickly evaluated so that the wide range of choices that occur over time can also be dealt with in an efficient and effective way.

B.4 The I.T. Framework Scope.

The I.T. Framework provides a mechanism for the analysis and definition of the technical infrastructure of an organisation. To achieve this it must provide for:-

- The definition of a common technical infrastructure for an organisation, across geographical, political and functional boundaries.

and then provide a way of delivering an effective mechanism for dealing with the differing profiles of different organisations. There is obviously great variation in the natures of different organisations and the systems bases that they require. This variation can, however, be related to three different infrastructure profiles.

The first applies when the organisation has a common set of requirements which can be delivered with one common set of solutions. A small organisation that required some marketing systems built around the basic support services such as payroll, personnel, finance etc. may fit this profile. The resultant technical infrastructure may well be based on a single processing platform, a single relational database with a 4GL development environment and packaged software solutions. It would therefore be possible to specify that the organisation has one common technical infrastructure for all applications.

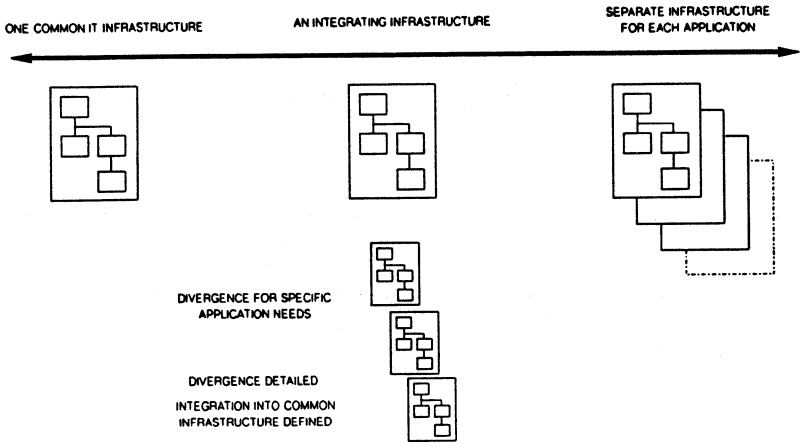


Fig.3 The scope of the I.T. Framework.

The second applies when the organisation has a need for integration and commonality between different applications, but also needs to provide the optimum solution for each of many different applications. For this profile there is a need for an integrating infrastructure. This provides a description of the infrastructure that is preferred because it addresses the integration needs across applications. When an application is developed it should reference the I.T. Framework that describes the integrating infrastructure. If that integrating infrastructure can support most of the functionality, then it should be adopted for that application. Where it is felt that key business functionality would be compromised by the adoption of the common infrastructure base, then a decision must be made on the best way forward. If it is decided that the business needs require a different technical base for this application then the divergence from the common technical base is documented and a mechanism devised to provide the best possible integration back to the common technical base. This approach is not dependent on the scale of an organisation, but can grow organically with the systems environment. Whilst it can produce a relatively complex technical structure across the organisation, this only occurs where the business requires it. The I.T. Framework provides the structure to document the logical steps taken, the reasons for them, and the implications so that the infrastructure base can be clearly understood.

The third applies when the organisation has no need for integration between different applications or has such widely differing requirements that each application is normally

fundamentally different from each other. In this situation there is still a need to carefully choose and describe the technical infrastructure for each application area. The result is a separate I.T. Framework for each of the application areas.

For a large organisation it may be that the second profile applies, but with a set of basic infrastructures each with their own divergence, across major political, geographical or functional divisions. The I.T. Framework provides the mechanism for documenting the chosen technical infrastructure for any of these profiles. The first step in producing the framework is to analyse the requirements and the current systems infrastructure so that the correct profile is identified. This process is not however only performed at the start of the definition of the infrastructure requirements. As the definition of the infrastructure progresses it may become apparent that there is either:-

- A different profile - for which the I.T. Framework should be adjusted.
- Or a fundamentally different set of requirements that give rise to additional I.T. Frameworks. This process would occur in much the same way that "marsupial" entities are derived when using the Jackson System Development analysis and design methodology.

The structure of the I.T. Framework enables it to, first cope with the profile of any organisation, and then to easily accommodate changes that arise as the framework is developed.

B.5 The Framework Structure.

The I.T. Framework is a statement of the information technology environment of an organisation or business unit. The framework supports the concept that although each component and element of the framework may be tackled separately they are also interdependent. The framework highlights the interdependency, provides a link between technical implementation and business goals, and details the technical solutions adopted.

The I.T. Framework comprises four mandatory sections :-

- Computing
- Communications
- Data Strategy
- Development Strategy

and a customisable set of sections utilised as appropriate. This set is defined on the

basis of each customer's specific needs. A commonly found set is:-

- User Interface
- Security Strategy
- Disaster Recovery
- Operating Strategy

Within each section there are four high level descriptions (these are taken from Earl 89):-

- Principles - these are the major design parameters of each of the infrastructure sections. They represent the essential requirements that each section should satisfy. For example, for computing a principle might be:-

  "Standards that allow the widest choice of software and hardware must be adopted".

- Policies - these are statements of how each technical element is to be delivered. For example, for computing a policy might be:-

  "Adopt open standards based upon X-Open and POSIX standards"

- Schema - this is a description, usually diagrammatic, of the proposed structure.

- Plans - Firm plans and goals for each element. For example, for computing a plan might be:-

  "Install open standards based systems for management information systems by the end of 1991".

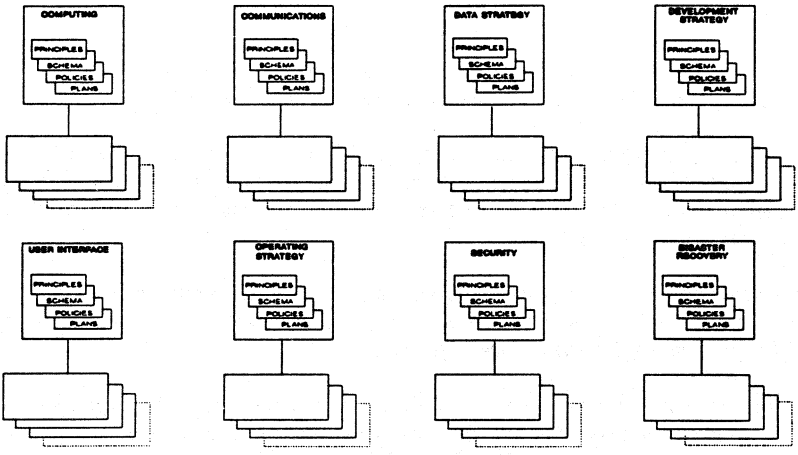


Fig.4 The I.T. Framework sections.

Then for each of the chosen sections the framework is split into a number of components:-

- Computing - Suppliers, Hardware Platform and Operating Systems.
- Communications - Internal and External.
- Data Strategy - Data Design, Databases and Database Administration.
- Development Strategy - as required.
- Others - as required.

Each of these components is then split into seven common elements

- Requirements - these are lower level statements for each of the system components that define the parameters which that component should follow. They are derived from the I.S. Strategy Planning process and refer to business needs. For example, for data design a requirement might be;

  Shared information should be controlled as a shared resource, not as if owned by one operational area.

- Technical Considerations - these are statements that refer to technical issues that need to be taken into consideration. For example, for data design a consideration

might be;

Shared information should be provided in such a way that the optimum number of computer systems can access it transparently.

- Risk - these are statements that identify the risk levels expected for the technical elements to be used. They are derived from the high level risk requirements defined during the I.S. Strategy Planning process. For example, for hardware platforms a risk assessment might be;

  The hardware base must provide a low risk platform for all operational systems. For other areas, such as the development strategy, the risk level expected may be much higher if leading edge systems are being produced.

- Candidate Solution Types - these are details of the possible technology solution types to meet the needs for each element. There may be one or many candidates. For example, for operating systems the candidates might include;

  Proprietary or "Open" operating systems.

- Candidate Solutions - these provide a more detailed analysis of the possible solution types, detailing the specific solutions available. It may be that only one of the candidate solution types is evaluated as a solution, or many may be evaluated. For example, for operating systems the candidate solutions might be;

  The candidate solutions are the UNIX based operating systems from major suppliers. The operating systems to be considered are HP-UX and SunOS.

- Chosen Solutions - these are descriptions of the chosen solution. For example, for the hardware platform the chosen solution might be;

  For large data processing systems HP9000 800 systems will be used with a mixture of multi-vendor IBM compatible PCs to be used for end user computing and client processing.

- Reasons for Choices - these are descriptions of the reasons for the choices that have been made. For example, for hardware platforms the reasons might be;

  The HP9000 800 series provides a high degree of price performance (better than $20,000 per TPS) coupled with wide software availability (over 4,000 applications currently available) , high reliability (MTBF figures of up to 20 years), excellent

scalability (a performance span of some 50X) and a rich set of peripherals (including Optical Disc, DAT, and CD-ROM). The IBM compatible PCs can be purchased according to the requirements and budgets of each operating unit. This provides them with flexibility of choice in terms of configuration, performance and price. Through our PC maintenance agreement multi-vendor PCs can be provided with full hardware support.

This structure enables the I.T. Framework to provide a clear and simple, but powerful description of the technical infrastructure of an organisation. Because of this clear and simple structure, it then becomes easy to identify where key decisions have been made and the reasons for those decisions. This enables a deep and common understanding to be produced that makes the evaluation of subsequent changes in business needs or technical options much easier to evaluate and implement.

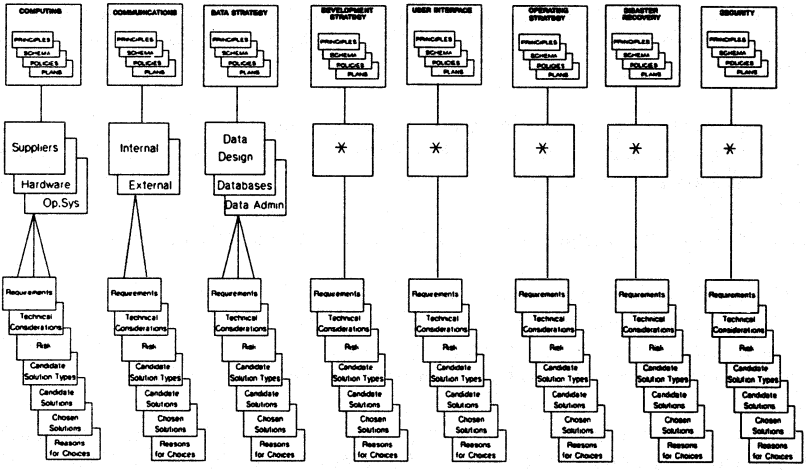


Fig.5 The Complete I.T. Framework Structure.

B.6 Ranking

The eight sections of the I.T. Framework provide for an effective description of the technical infrastructure of an organisation. A key element in determining how to transform the requirements for each section into products and services, is the relative

ranking of the sections. This ranking process enables the requirements to be positioned relative to all of the other requirements and enables an infrastructure to be defined that is properly integrated rather than composed of eight isolated island that only refer to their parochial requirements. For example, if Data Strategy is deemed to be the number one ranked issue, then the solutions developed for each of the other sections must follow on from any solutions identified for Data Strategy and not compromise the higher level decision. The same follows for the relative position of each section as it is evaluated. By using this ranking approach a solution can be developed that provides optimal integration between the conflicting needs of different sections, based on the initially identified relative business need that drives each section.

B.7 Deriving the I.T. Framework

The production of the I.T. Framework as part of the I.T. Strategy Planning process is, in terms of the individual steps, straight forward. As mentioned in section B.1 the framework is produced as part of a process that follows on from previous analysis that defined:-

- The business needs that the organisation should be addressing.
- The way that those needs could be addressed through the use of information systems based solutions.

The initial information required for the I.T. Framework to be produced is:-

- The high level functional needs.
- The high level view of the infrastructure requirements.
- The high level assessment of appropriate risk levels.
- The cost/benefit profile that the infrastructure is expected to deliver to.

This information provides the base upon which the decision process in producing the I.T. Framework is founded. The I.T. specialists provide an addition to this base, by then defining the technical considerations and risks that need to be generally borne in mind. This information is laid out in a simple logical fashion under the headings of, Requirements, Technical Considerations and Risk.

If the first assessment of technical considerations highlights that there will be some problems in meeting the requirements as defined, then these should be immediately addressed. The simple logical steps involved in developing the I.T. Framework make these issues fairly clear.

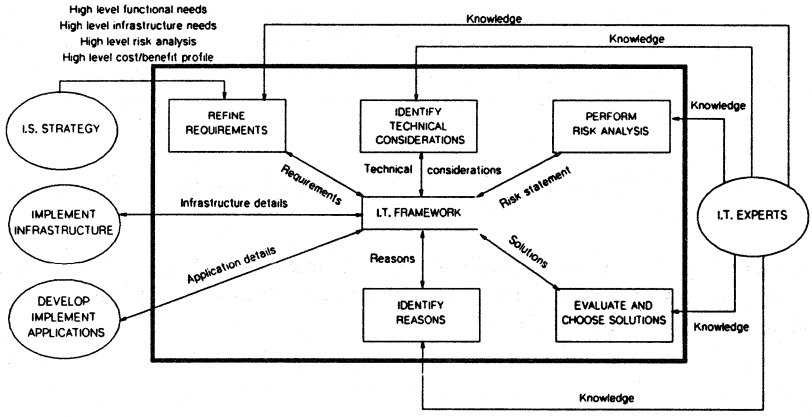


Fig.6 Deriving the I.T. Framework.

The sections should then be ranked in order of the importance of each section as derived from the requirements and the higher level business needs. If the ranking does not fall out fairly easily then a deeper analysis of the business needs should be commenced to provide enough detail to generate a meaningful ranking statement.

The next step is to look for the potential solution types that could address the requirements. Again it is important to make sure that any potential solution types and consequent solutions do address the requirements taking all the attribute requirements, like cost, time, risk, performance and quality into consideration. If at this point there are some major issues in the capability of potential solutions to meet the requirements the information needs to be feed back to groups making the directing business decisions. If there are no solutions available then the requirements need to be re-evaluated in that light. If there are solutions available, but they do not meet the required attribute specifications, then again the business needs verses the benefits must be re-evaluated. In this way the process of technology evaluation can be directly related to the analysis of business requirement. Where the desired cost/benefit profile cannot be delivered the issues can be raised immediately and addressed. This avoids the commonly found situation of the I.T. department being told to produce a systems solution for the organisation's infrastructure needs, with no mechanism in place to test the practical viability of that solution, and enables the resolution of problems as they arise during the planning process, rather than after implementation.

When the possible solution types and solutions have been evaluated the final choice can be made and the reasons for the choice described. This definition of reasons is key as it provides the only effective way of being able to evaluate the chosen set of infrastructure solutions and the effects of future changes on the appropriateness of the choices.

C. COMMON PROBLEMS THAT THE I.T. FRAMEWORK HELPS TO ADDRESS

The clear, simple step by step structure of the I.T. Framework enables it to be used in a way that deals with some of the common problems encountered when trying to implement strategies in the real world.

C.1 Hidden Agendas.

For any situation a split can be defined between the aspects that relate to the emotional understanding and effects, and the underlying process that is being followed. These two dimensions are crucial to the successful management of change and both need to be approached with care. Underlying the analysis, definition and implementation of a new systems infrastructure should be a process that takes the requirements and clearly details the resulting steps to a solution. When that process and the resulting conclusions become affected by hidden agendas with conflicting goals, as they almost invariably will, then it becomes very difficult to identify the logical requirements that the process addressed and the reasons for the choice of the solutions. Both the emotional and the logical process aspects of the exercise need to be dealt with in such a way that they are both addressed, but that each one clouds the other as little as possible. The emotionally derived hidden agenda issues that might be involved are:-

- Resistance to change.
- Holding on to a power base.
- Political manoeuvring.
- Loss of status.
- Realisation that skills are out of date.

The I.T. Framework provides a way of highlighting all issues through its simple, clear, step by step approach. In doing this it can identify the type of requirements that are being addressed. It is important to be able to identify which allowances and/or policies are being adopted and whether they are aimed at solving the emotional or logical issues needed to deliver the most effective solution. Without a clear definition of all the

requirements many systems developments flounder.

C.2 Confusion of Solutions with Requirements.

Tom Gilb is a well known "guru" in the software development world and has written widely on what he considers to be the main problems with software development. Three quotes from his 1988 book "Principles of Software Engineering Management " highlight these concerns. Among many principles he defines are the following:-

. The principle of fuzzy targets:-

Projects without clear goals will not achieve their goals clearly. (You can't hit a bullseye if you don't know where the target is)

. The principle of the separation of ends and means:-

Avoid mentioning solutions in your goal statements.

. The principle of the obvious:-

. *"Obvious things" which "everybody knows" cannot be left to take care of themselves.*

The I.T. Framework addresses these principles directly. It provides clear statements of requirements for infrastructure needs. It requires the documentation of each "obvious step" so that the decision processes can be followed clearly. By separating the requirements from the solutions and documenting the derivation of the solutions, it provides a clear view of how well any solutions meet the requirements and where solutions have wrongly become an end in themselves.

C.3 Lack of Communication Between the User Community and System Builders.

In the understanding and communication of any process there are some fundamental process control dimensions of the information transaction that must be addressed:-

. Authority - the final say in what is done.
. Responsibility - the job of making it happen.
. Knowledge - an understanding of the processes.
. Skills - the ability to make the processes work in practice.

One of the major problems in effectively delivering any systems element is the lack of a clear definition of who can provide the correct information for each of these four levels of control. By making the infrastructure requirements and the potential solutions explicit it becomes clearer which requirements refer to which level of of control. By providing this clarity the I.T. Framework provides the raw material to assist in the amelioration of the problems usually caused by this confusion of levels of control.

C.4 The Natural Dynamic and Chaotic Nature of Systems.

Information systems and the supporting technical infrastructures exist for one reason. To enable a business or a process to perform in a more efficient and effective way such that the investment in the information systems generates an acceptable rate of return. The nature of business processes over the last few years has dramatically changed to a highly dynamic and changeable one (Anniss 1990). Some business processes remain static, but many are being reviewed and amended at least once or twice a year. In such an environment it is imperative that the computer systems developed to support the business needs are able to deal with this constant change in addition to the constant technological change that has led to the release of new products from computer companies on a six month cycle.

The I.T. Framework greatly assists in controlling this change by:-

- Making each step in the decision process clear.
- Enabling the effects of change to be easily assessed against the current infrastructure.
- Providing a focus on key infrastructure issues such as "Openness" and "Flexibility".

D. OPEN SOLUTIONS

Open solutions have two fundamental dimensions, open platforms and open software. The move towards standards based open systems during the late 1980s and early 1990s has in general been based on the development of open platforms. This has affected both the hardware platforms and the application software.

- The open platform base has been addressed by the provision of standards based operating systems based on POSIX and UNIX that enable a wide range of hardware platforms to be the base of a common operating environment.
- The application software developers and the development tools providers have used

the wide availability of this common operating environment to provide sets of tools and solutions that can be moved across different hardware bases with relative ease.

This has resulted in a high degree of vendor independence in relation to the hardware suppliers such that any open systems based solution can be moved between the chosen hardware platform and an alternative with relative ease. In practice this degree of vendor independence is not available for the software component of the open systems based solution. Whilst open systems provide a wide initial choice of software applications and tools, once the software solution is chosen it will often be difficult, and in many cases impossible, to move to an alternative software solution without a major re-development of the software.

The main standards body directing these Open Systems issues is X/Open, whose mission is to "Facilitate, guide and manage the process of developing a *Common Application Environment,* or CAE, based on de-facto and industry standards". The foundation of this CAE is the POSIX interface definition. The CAE attempts to extend POSIX into a complete operating environment by addressing additional requirements such as data management, integration of applications, data communications, high level languages and user interfaces. The current definition of this software environment is in the reference document called X/Open Portability Guide 3, or XPG3.

In practice while XPG3 defines a portable environment for the development of software, its components are very restricted. Most software tools generally available in the market place provide facilities that extend much further than those defined in XPG3, which has led to this lack of "real" openness in software solutions. Current work by X/Open on common database access, based on the SQL Access group, and on a more complete user interface definition, has not yet produced the next common standards definition, which will be labelled XPG4.

A way of clearly defining the open software issues is shown in Fig.7. This identifies three levels of "openness" in software that relate directly to the needs that specific applications should be addressing.

. Level 1 is characterised by the use of platform portable proprietary software products. In general the set of products provide

  Advantages:-

  . Platform independence
  . Access to advanced, non standard, proprietary features such as stored procedures, object orientation.

- Flexibility of integrated 4GL type environments
- Faster development in the short term

Disadvantages:-

- Software vendor dependence.
- Tie in to proprietary features.
- If 4GL products used, potential limited extendibility and capability.
- Potential lock out of new technologies and techniques.
- Potential "cul-de-sac" development in the long term.

- Level 2 is a combination of levels 1 and 3, and occurs when the business needs have identified that particular software components must be open, while other components do not need to be open. In general the set of products provide:-

Advantages:-

- Mix and match products for functionality and openness where appropriate. Can have openness in particular areas such as the GUI or database access.
- Choice of the degree of software vendor lock in.
- Systems developed with a focus on both immediate functionality and future developments, as appropriate.

Disadvantages:-

- Still the potential for software vendor and feature dependence.
- Need to understand in detail the implications of the different choices.

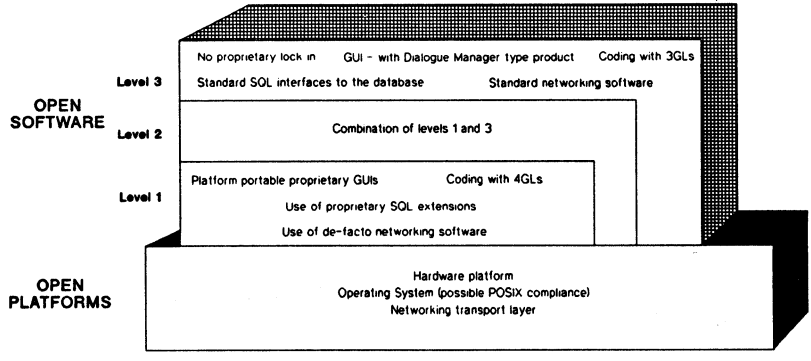


Fig.7 Open Platforms and Open Software

. Level 3 is characterised by a need to build software that provides for a long life and flexibility. In general the set of products provide:-

Advantages:-

. Platform and software vendor independence.
. Openness to new technology.
. Faster more flexible development in the long term.

Disadvantages:-

. Longer development time in the short term.
. Not able to use useful proprietary features.
. Requires in depth knowledge of software architectural issues.

Given this open software component, Fig.8 highlights one of the key issues. To enable new components to be assimilated as painlessly as possible into a system structure, the links between them must be open and normally accessed by an Application Programming Interface, or API. If the components such as language (3GL or 4GL), user interface and database management are uncoupled in this way then the ease with which each part of the systems can be replaced as new products become available or business needs change is greatly enhanced.

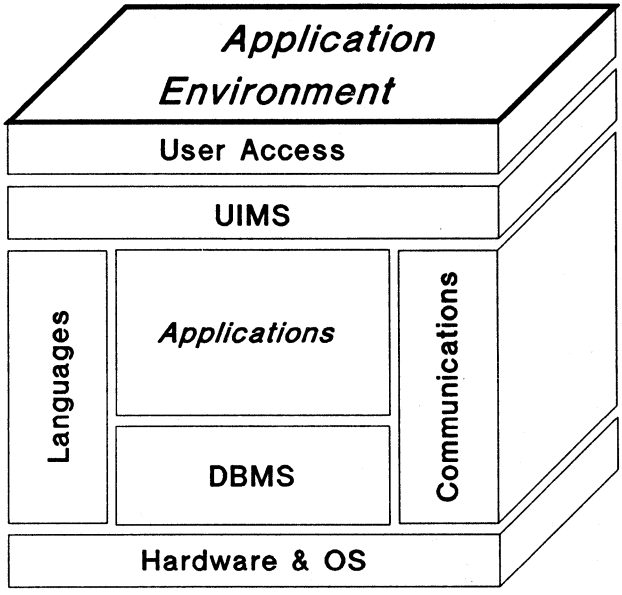


Fig.8 The Structure of Open Software

The I.T. Framework provides a structure within which these open platform and open software issues can be clearly positioned. It describes how the requirements lead to the choice of one or both of these approaches as future fundamental infrastructure drivers. The management of this open solutions based environment is complex because of the wide range of choices available and the long term implications of infrastructure solutions. The open platform choice is more obvious and direct as it refers in the main to the provision of a common operating environment, probably based on POSIX standards. The open software choice is much more problematic. The above description of the advantages and disadvantages of adopting different approaches highlights the major trade offs that need to be made in terms of, the development of current systems to meet immediate needs, and the openness and flexibility of systems to adapt to the future and address such problems as the growing application backlog. To manage this issue it is crucial to provide a mechanism to enable the definition and evaluation of the broad view across the business needs of the organisation, and to detail the reasons for the solutions that have been chosen. The I.T. Framework provides this mechanism by making each choice clear and showing how it was derived. It further enables the

solutions to be modified and adapted over time as business needs and open software options change through continual revisiting of the choices made and analysis of the reasons in terms of developing and changing requirements.

E. REFERENCES

1. Management Strategies for Information Technology
   Michael J. Earl -Prentice Hall 1989
2. Principles of Software Engineering Management. Tom
   Gilb -Addison Wesley 1988
3. Evolution, the Key to Successful Software Engineering in the 1990s.
   Mike Anniss-Proceedings of the UK HP CUA July 1990.

Paper Number 5167

Performance Is A Dirty Word Around Here

or,

"How to get your act together using Capacity Management"

Doug McBride
Hewlett-Packard Company
Performance Technology Center
Roseville, CA

Introduction

The word "performance" when used in a sales or marketing situation, typically implies the positive; faster cars, clearer phones, more "powerful" computers. When the word "performance" is used in a **service** context, especially in regards to computers and their use in data processing, it's rarely used in the positive context. When was the last time you had someone run up and exclaim "Gosh! The computer sure is performing well today, isn't it?"?

The term that usually precedes the word performance relative to data processing is "poor", or "lousy", or even "unacceptable". The one word that comes to my mind which always seems to follow performance is "problem". Hence we get to the crux of the paper title **"Performance is a dirty word around here"**. But, before we get started discussing solutions to performance problems, I'd like to talk a little about technologies in general and how they are perceived by those who use or are affected by them.

A Simplified Technology Lifecycle Model

As an industry or technology matures, users of that technology begin to expect levels of service beyond that of a new, or fairly new technology. Studies have shown there's a small percentage of a user population that will typically try "new" types of technology or services. Their expectations are set accordingly, they are willing to risk problems and failures to stay ahead of their peers and competitors, and potentially reap the benefits (risk/reward) new technology can bring them. As a technology matures and becomes stable, the more conservative segment of a user community will see it as being cost effective enough to incorporate into their businesses or daily lives (or it gets incorporated into products they want to consume as a matter of course). Expectations also rise as to the level of service provided using that technology, due to it being viewed as stable and reliable.

As a technology becomes pervasive, it also has a tendency to become transparent - because if it wasn't, it probably wouldn't have become pervasive in the first place (it appears the

VCR may be one of the glaring exceptions to this rule, however). Pervasive technologies provide levels of service which are usually taken for granted, are almost always available, and if lost can/are restored quickly. Pervasive technologies are used by business to help promote growth, and provide stable platforms for new ventures or the streamlining of current operations to become more efficient and cost effective.

An example of a pervasive technology is the telephone system. We take it for granted that even on stormy nights when the power is off, we can go to the phone and find a dial tone. We always expect when we dial a number, the call will get through to the other end. We, for the most part, don't care about the incredible complexity of the system itself, how it really works, or why it works that way. We just want to be able to communicate from our homes, businesses, cars, golf carts, etc., and the heck with the details. When that technology is not available or doesn't work as expected, it has a pronounced effect on how we go about our business or our lives.

It's time to acknowledge that the use of computers for general data processing tasks is not only mature (at age 30 one would expect it to be relatively mature and shouldn't be living at home anymore), but is begging to and must, at some point become pervasive. (I say for "general data processing tasks" because the processing element of the computer has already become pervasive, has allowed some other related technology to become pervasive, or has sufficiently increased the quality of older technologies such that they've become more stable and reliable (cars, consumer electronics, etc.).)

In today's business environment with increased competition in many different sectors, businesses must run efficiently, and accommodate growth as effectively as possible. Data processing has become a critical part of most business operations and is used extensively to support the mission of the company. More and more manual systems are falling by the wayside, with new opportunities being realized that would be impossible without the use of a computer system to automate the collection, storage, and retrieval of information, or to automate functions which once required the use of human labor to accomplish. The ability for business to be able to leverage off of data processing for growth is critical, and the technology must be able to keep up with business growth. The data processing organization should take an aggressive position in bringing technology to the business to stimulate and lead in areas of growth.

How do we go about helping with this transition from "mature" to "pervasive" technology with regard to data processing systems? We successfully transitioned the technology from the "new" to the "mature" stage over the last 20 - 30 years by providing solutions to problems which before the application of computer systems were manual. We then made those solutions available for use on a regular and reliable basis. The solutions were accepted in day-to-day business practice, and the computer became more of a normal tool of business, rather than problem-prone, unreliable, and somewhat scary behemoth.

The transition from mature to pervasive status, is much more **qualitative** than **quantitative** however. As you recall from previous paragraphs, pervasive technologies are essentially transparent, and if they go awry they are usually repaired and available in minimum time. We all know of the quantum leaps we've made in the last few years in terms of data

processing system reliability and mean time to repair. But what of **performance**? When we have a performance problem, the technology can no longer be transparent. It no longer is a reliable, predictable platform from which to launch and support new business ventures. Merely supporting current business environments becomes a chore when your data processing system can't get the work done in a timely fashion. Upper management within the business are "aware" of data processing in their business, but not necessarily in the ideal context.

Consequently, it is necessary to be able to put in place a data processing environment that meets the functional needs of the user base, while at the same time, performing at a level that meets the user community's expectations for performance. Only then will data processing be truly integrated into the business and achieve the transparency required of a pervasive technology.

Why Are There Performance Problems?

Significant material has been written exploring the details of why there are performance problems with data processing systems, from expectation/perception situations to incredibly detailed diatribes about operating system algorithms and their effects on system performance. I would just as soon keep it simple, and pose that computer system performance, much like the performance of any other system, **is a function of the capacity of that system to handle the requests made on it, to do the work its supposed to do.** Wait! Wait! I can hear you claiming this is a cop-out and we're really not talking about performance here. The obvious couldn't have been stated better, right? Thanks for the complement, but read on.

Many times when we try and address the data processing performance problem, it's easy to get bogged down in the details too early in the cycle. Details are fine when we're going looking for solutions. However, if we can acknowledge that for any given system workload, there is a set of resources that are needed to meet system performance parameters, we're well on the road to a better way to manage our systems. It allows us to **prevent** problems from occurring in the first place, and with the absence of problems, we're that much closer to providing the "pervasive" technology our businesses need to be competitive and grow.

What do we really mean by "capacity" when talking about data processing systems? There are many definitions, again depending on how many nits you'd like to pick. But simply put, **the capacity of a data processing system can be measured in the amount of useful work that can be accomplished at the requisite service levels needed to support the entity using the system.**

The physical parts of a data processing system associated with capacities are: the Central Processing Unit (CPU) which actually does most of the processing work; the size of the CPU's Main and Cache Memories, which hold programs and data the CPU works upon; and the ability of the system to get things in and out of the CPU and Memories, or what we normally refer to a the I/O Subsystem.

What happens when the work we want to get done exceeds one or more of the capacities of our data processing system and why does the system then seem to "slow down"? Just like any other service center that has a finite ability to do some work or perform some service in a given amount of time, when the service center is being used to its full capacity, the entities that require service from that service center (in our case, the CPU, Memories, or I/O Subsystem for example) must then wait their turn in line to get service. This is normally referred to as "queueing" (standing in line). The study of the behavior of queues and the effect that behavior has on those receiving service from the center being waited upon is called **Queueing Theory**, and has quite a following in the planning and capacity management area, as you can well imagine.

So in essence, our data processing system doesn't really "slow down", as much as it just can service so many requests over time. When the work demanded of the system exceeds its capacity to deliver the work, the queues get longer. Assuming an evenly distributed service time from the service center that's in demand, you can extrapolate that the amount of time spent in line is equal to the number of people in front of you, times the service time at the service center. A key notion here as well is that of "specified period of time". As we all know, the work usually gets done - it's just a matter of when and what effect waiting has on the entity needing to get the work done.

A good analogy used when talking about queueing to a service center is that of a bank. The bank has a certain number of tellers (service centers) who deliver banking services to their customers (us). When there are enough tellers to handle the number of customers arriving to do banking, the wait is either non-existent or very short. Consequently from the standpoint of the bank's customers, they're getting "good" performance and they focus on the quality of the actual service they receive. But during lunch hour, if the bank keeps the same number of tellers on duty that it uses for the early morning hours (assuming they're open), the lunch rush exceeds the capability of the tellers to service all customers in a "reasonable" amount of time, causing folks to wait in line - sometimes for an inordinate amount of time. Performance is now "poor". Even if the quality of service from the tellers (service centers) is quite good, and taken care of in a timely manner, the focus will tend to be on how long the wait was to get served in the first place.

We all know the best way to solve the problem from a customer's standpoint is to add more tellers (service centers) so more work can be done all at once (in parallel). But other options available to the bank manager include limiting the number of customers that can be in the bank at any one time to be standing in line (scheduling), or even innovative ways of standing in line (queuing disciplines). The tradeoff the bank manager has to face is how many tellers (capacity) he/she must have to make sure their loyal customers don't start doing business down the street because it takes so long to get a banking transaction taken care of in his/her bank. Too few tellers and customers are unhappy and may go elsewhere. Too many tellers and the bank's profits start to decline, making shareholders unhappy and jeopardizing the managers job. So we have a delicate tradeoff between too much capacity to handle peaks, and too little capacity when the rush comes. I guess we can all agree that's why managers get paid "the big bucks" to try and sort the problem out.

Taking this analogy and applying to data processing systems, we may have several applications (workloads) which make up our total system workload. We have a number of users of those applications that expect a certain level of service (response, throughput) of the system. If the system as a whole is sized correctly (sufficient capacity), and there are no logical bottlenecks which exist as a function of any of the applications, the users of the system get adequate service (short lines). If for some reason the demands placed upon the system are increased to the point where queueing starts to occur (an inordinate increase in the number of users of one application or another, or a new application is added without regard to current system capacities), the system begins to "perform poorly" (we stand in line for quite some time). When this happens, the allotted work cannot be accomplished in the timeframe it is supposed to be done, users are unhappy, business isn't getting done, and perhaps the data processing manager is asked if he/she is really doing his/her job.

Capacity Management

Returning to the discussion of technology lifecycles, it's important to note that in parallel with the transition to "transparency", comes a more proactive and scientific approach to the management of the technology. As we've discussed, having sufficient system capacity is key to preventing performance problems, which in turn leads to what is perceived as an effectively managed system (assuming application functionality is available and proper).

It's rare to find any major system in a business environment whose management is not concerned about the proper handling of that system's capacities. Could you imagine how long a manufacturing shop floor manager would stay employed if he/she couldn't ensure proper machine tool capacity to get all the widgets built to meet current orders? Or conversely, having significant idle time on many very expensive machines? Managing the capacities of large capital systems in business is a given in order to ensure a return on the investment, and to stay competitive. No one says it's easy, but it is a business fact of life. And, the more effectively it can be done, the more profitable the business will be in the long term.

Now it's time to start applying those same capacity management concepts to our data processing systems. Businesses that can effectively manage the capacity of their computer systems will be more competitive, and be better able to use the power these systems to help grow the business. The question that goes begging is one of implementation specifics. Is is possible to manage the capacity of a data processing system in relation to applied workloads, and if so, how is it done?

The techniques for managing data processing system capacity have been known for many years. They have been practiced primarily in the mainframe world where large, centralized data processing organizations have supplied high levels of service to application groups in their companies. With several hundred to sometimes thousands of users depending upon large systems for service (and the underlying success of the business as well), and the cost of bringing additional capacity on line when needed being in the several million dollar range, it's been imperative that data processing managers practice capacity management rigorously. Consequently the tools and processes to support the management of system

capacity are available and in wide use. An international organization even exists (the *Computer Measurement Group* - **CMG**) whose charter is to promulgate and encourage information exchange on the subject of computer system performance evaluation and capacity management.

How does one go about managing the capacity of their data processing system then? Why don't more data processing managers practice capacity management techniques? I believe many data processing managers are not familiar with the notion that computer system capacity can indeed be managed, and consequently have given in to the notion that the associated performance problems are a fact of life. Therefore they are mostly in a "reactive" management (an oxymoron?) mode. They are unsure what effect, if any, adding workloads, or increasing workload intensity will have on their systems.

In the next few paragraphs, I'll introduce some management techniques and procedures that can be used to take and maintain control of your data processing systems - allowing a chance to transition into a "pervasive" technology role within your company, and your ability to realize the associated benefits thereof.

Taking Control

In order to manage the capacity of your data processing system, you need to know two things: **1) What the current capacity of your data processing systems is and how it is being used**; and **2) How you intend to use that capacity in the future to accommodate growth or change within the business**. In many cases, the perceived inability to address item 1 has stopped managers from attempting to manage their system capacity. If there is no obvious way to determine how current system capacity is being used, how can you even start to plan?

Good question! If the mainframe folks have been doing it for years, how have they managed? As I mentioned before, they have tools (typically software written to help in the monitoring and management process) that help them do it. Similar tools are also available for use in the Hewlett-Packard computer environment. These tools are used to collect system utilization data over time, with the ability to attribute back to specific applications the amount of system capacity required to run them, and what the results the users are seeing (response times, transaction throughputs, etc.) as the application runs on the data processing system. By using these tools, the manager and his/her staff have the ability to determine how much system capacity remains for growth, and when specific applications have grown beyond the ability of the system to handle them. There also exist tools which allow the ability to forecast into the future what additional capacity needs to be brought on line in what time frame to handle anticipated growth.

Tools to monitor and forecast system capacity are certainly a necessity. Proper capacity management cannot even be imaged without them. But tools alone do not solve the problem. A framework for effectively documenting how the system is used by the different application user groups is necessary in order to understand what their expectations are of the data processing system. Remember, if it weren't for the users of the system, why would we need the system in the first place? Again, in order to achieve the transparency

associated with the pervasive technologies, you have to understand the levels of service the users expect, and maintain those service levels even in times of rapid change or growth by bringing additional capacity online **before** performance problems occur.

One such way of getting control over this situation is through the use of Service Level Agreements (SLAs). A SLA is a document used to isolate the needs of specific user groups of the data processing system, what service levels (response, throughput, quality, availability, etc.) the user groups can expect, and what the data processing organization can expect from the user groups to help them meet the specified service levels.

Many informative papers have been written on the subject, two of which are [MILLER87] and [MCBRIDE90]. As a matter of fact, there is an incredibly rich set of performance and capacity management references available through the *Computer Measurement Group*. (I've included their address and telephone number in the references section in case you're interested in getting more information, or attending their annual international conference held in the second week of December at various locations throughout the U.S.) The SLA allows the data processing organization to fully understand and document what the user community expects of them, and to extrapolate those needs into metrics for use in overall capacity management of all systems involved in delivering service to users.

Once a SLA is established for a particular application user group, a set of Service Level Objectives (SLOs) can be established for internal use within the data processing organization. The SLO is a lower level document used internally by the data processing staff to map computer metrics to user services levels, and to establish a measurement and monitoring function for the services levels [MACKINNON86], [MARSH89]. By monitoring current service levels against contracted service levels, a satisfaction index can be derived which should be an indicator is how well data processing is doing its job, and how it is meeting the business goals of the organization within which it exists.

By taking the information gathered in the previous steps to measure and monitor system service levels, an incredibly useful byproduct emerges. Usually we find we have the ability to attribute the amount of data processing system resources needed to support various types of user transactions. (If we didn't have this information from our tools or some other extrapolation technique, it would be difficult to report on current service levels in a fashion meaningful to the user groups contracting for service.) Given we have this information, we can now do forecasting based upon the growth within a specific application, and usually do so in regards to how the larger business organization is growing (often referred to as **Natural Business Units** or **NBUs**). Doesn't it make more sense to ask how much more computer capacity we'll need if sales quadruple over the next nine months, as opposed to some other lower level measure which is meaningless to the user community? In this fashion, the **customers** of data processing services become allies in doing capacity management, and we can use them as a prime information source in determining when additional capacity will be needed.

Finally, it's possible to use mathematical models of computer systems to forecast system capacity even when the characteristics of a new workload, or combination of workloads are unknown to you. Professional services are available from consulting firms (including

Hewlett-Packard) which allow total "what if" scenarios to be developed with known and unknown workload characteristics, with the output of the exercise being a set of capacity metrics which can be applied to machine and peripheral complement sizing, for both the short and long term.

Summary

The intent of this paper was not necessarily to provide a cookbook approach to managing the capacity of your data processing systems - that effort is still in the works. However, it is necessary to acknowledge that the act of capacity planning and effective data processing system management is a reality before you can embrace it. If we're to expect data processing to become a pervasive technology and bring to the business organization within which it exists the associated benefits of a pervasive technology, capacity management is a must. It is the **qualitative** difference between keeping a system up and available, and ensuring sufficient capacity to make performance problems a thing of the past, with transparent service to users the norm.

Once the need and capability to do capacity management is acknowledged, then plans can be put in place to do so. As data processing technology becomes pervasive across the myriad of industries it supports, so will the management mechanisms and techniques it took to get it there. I encourage and challenge you to explore the concepts presented in this paper, and to look outside of your standard sources of information for more material on the subject. You might also look outside of your organization to professional services providers for help in getting started managing the capacity of your data processing systems in relationship to the business goals of your organization. Many different levels of service are available, from initial data on the subject, to complete outsourcing of the SLA process and associated capacity management function. Regardless of the mechanics of implementation, capacity management is, and will be even more so, an important and necessary function within your business as we move into the competitive environment of the 90's and beyond.

References

[MILLER87] Miller, George W. (Bill), "*Service Level Agreements: Good Fences Make Good Neighbors*", Conference Proceedings of the Computer Measurement Group (December 1987), pp. 553-557.

[MCBRIDE90] McBride, Doug, "*Service Level Agreements: A Path to Effective System Management*", HP Professional, August 1990.

[MACKINNON86] MacKinnin, Douglas, "*Service Goals: Who Needs Them?*", Conference Proceedings of the Computer Measurement Group, (December 1986), pp. 618-622.

[MARSH89] Marsh, Robert E., "*Measuring and Reporting User Impact of Poor Performance*", Conference Proceedings of the Computer Measurement Group (December 1989), pp. 1011-1017.

For more information on the subjects of Computer Capacity Management, Service Level Management, System Modeling, (all from the standpoint of both the technician and the manager), contact the *Computer Measurement Group* at:

The Computer Measurement Group, Inc.
401 N. Michigan Ave.
Chicago, IL 60611-4267
(312) 527-6652

DURRE PLACEMENT SERVICES

SURVIVING WITH THE SIMPSONS

(And I thought these guys were just TV Cartoon Characters)

THE SIMPSONS. Everyone in America knows who they are. From bit parts on the Tracey Ullman show, these unlikely heroes have become world celebrities. Now everything from T-Shirts to underwear bears their pictures...and there seems to be no end in sight.

There probably has never been such an unlikely hit - a Cartoon series on a relatively small TV network (Fox), competing in its time slot against one of the top shows (Cosbys) on a major network...And the characters are not typical hero types - no superhuman powers, unattractive (if not downright ugly). Their personalities are much the same - not cute but just plain disgusting for the most part. There is not one wholesome, "together" role model among them. So why have "The Simpsons" become such a mega-hit? What is their appeal?

Perhaps it is precisely because they are more "human" than most "real" characters we encounter on TV. They are in fact mainstream American personalities. Bart says & does what we are often thinking. Homer's life is the embodiment of Murphy's laws fleshed out in real life experiences. They show us the best and worst in ourselves, and since they are cartoons, we can laugh at them and never admit that the joke is really on us.

But this session is not primarily a review of a popular TV series. It is a help session focusing primarily on people management. Truth is, most of us know at least one of the Simpsons - up close and personal. Many of us work beside or around them every day. If you are a manager, you probably have at least one of them on your staff. At the risk of blowing the entire premise of this session, I want to see where we are as a group...

* How many here have supervisory responsibility for at least one other than yourself in your present position?
* How many of you are managers with 4 or more employees reporting to you?
* Of that group, how many of you already know that you have at least one of the Simpsons working for you now?
* I ought to ask a more dangerous question - How many Simpsons do we have here today?

In truth, I couldn't ask for a show of hands of Homers. If you are a real Homer, you probably don't know it. I also couldn't ask for a show of hands of Barts. If you are a hardcore Bart, you're likely opposed to anything that smacks of learning or self-improvement so you wouldn't be here (unless you brought a slingshot or plan to throw a lot of spitwads and insults.)

However, the value of this session will depend on your willingness to deal with both lines of questioning. 1) Who are

you as manager, and 2) Who are the people you work with and supervise?

Realities of Management:
- According to recent surveys, managers rank near the bottom in terms of admiration & credibility. **They are the Rodney Dangerfields of the career set. They "get no respect"**...(Yet when I ask people about their future goals and position objective, they inevitably say, "I want a position leading into management." (Figure that one out)

- **Management is usually thrust upon a person, untrained & often unskilled for the position.** Hiring decisions for managers are typically based on 2 factors: 1) tenure, and 2) performance in non-management functions. In DP, a gifted programmer can distinguish himself in programming so that he is promoted into a management position where he will never again use his gift of programming. All his technical training and experience will earn for him a position where his technical expertise has little or no affect on his ability to do the job.

Perhaps that explains why managers often get poor reviews and a general lack of respect. While this session will not correct either of the above problems, perhaps it can provide a small start toward helping you become as adept in dealing with your people resources as you likely are with your technical resources.

I must begin by sharing with you some presuppositions that form the foundation for this session:

PRESUPPOSITION #1: PEOPLE DO NOT HATE WORK. WORK IS AS NATURAL AS REST OR PLAY. MOST PEOPLE LIKE TO WORK.

My personal conviction is that laziness & lack of motivation are for the most part mis-conceptions. Every person I have ever met worked hard at something. One of the most highly motivated, hardest-working people I ever met was a professional con artist. He took me for $85.00. It took him 1 1/2 days to do it, but he did it. He worked harder for that $85 than most employees I have supervised have worked for significantly more than that amount. His problem was not a lack of motivation, but that his motivation was sadly misdirected.

Back to the Simpsons: Bart is a perfect example of the same problem. He is a poor student, a behavioral problem. In fact his teacher's greatest nightmare is that there may be more where he comes from. "UNDERACHIEVER...and PROUD of it!" Yet is he unmotivated? Hardly. Bart stays up nights thinking up ways to disrupt class. His goal in life is to drive a certain bartender crazy with his prank calls (he's succeeding). His creative juices are flowing...just flowing in the wrong direction. He is driven to achieve, and he will reach every goal he sets for himself. The challenge with a Bart is not to motivate him, (he is motivated!) but to help him set better goals and to channel his energies toward achieving them.

Leads to my 2nd presupposition:

PRESUPPOSITION #2: PEOPLE ARE MOTIVATED TO ACHIEVE AND WILL DRIVE THEMSELVES BETTER THAN WE CAN DRIVE THEM, SO LONG AS THEY ARE COMMITTED TO OUR MUTUAL OBJECTIVE.

Douglas McGregor, The Human Side of Enterprise, says that most companies wrongly operate under some false assumptions:
He calls this his "Theory X":

1. People hate work.
2. They have to be driven and threatened with punishment to get them to work toward organizational objectives.
3. They like security, aren't ambitious, want to be told what to do, & dislike responsibility.

Because of these assumptions, the traditional approach to management has been similar to an old west cattle drive with "cowboy" type managers to keep the herd in line. Uniform office hours, unilateral promotions, hourly wage vs. production wage, on-the-job control through threats of punishment, etc. are examples of those assumptions at work. Even the rewards offered: medical & death benefits, pensions, vacations, bowling leagues & baseball teams -none of which can be enjoyed on the job. You've got to leave work, get sick, retire, or die to realize the benefits of employment. No wonder people aren't motivated or having fun on the job.

The problem isn't that people hate work, it is that we have misunderstood what motivates people. People's NEEDS form their value hierarchy that is internal, not external. Motivators must be attached to those "needs".

1. **Physical (I can't breathe)**
2. **Safety (How can I protect myself from..?)**
3. **Social (I want to belong)**
4. **Ego (Tell me how terrific I am)**
5. **Development (Gee, I'm better than I was last year.)**

People are totally motivated by each level of need in that order - until that need is satisfied. Companies have typically recognized #1 & #2 and frequently #3. The key then in day to day motivation and management will be relating our mutual objectives to needs #4 & #5. Employees will buy into our mutual objectives & commit themselves to them as long as they see those objectives relating to reaching those other personal "needs." The trouble is that these needs are individual & personal and will vary from one person to the next. (Which leads to Presupposition #3:)

PRESUPPOSITION #3: PEOPLE ARE DIFFERENT. THEREFORE TO BE EFFECTIVE, MANAGEMENT MUST TAKE INTO ACCOUNT THOSE INDIVIDUAL DIFFERENCES.

Let me illustrate the problem. If motivation is tied closely to personal priorities, then what is the #1 priority or the prime motivator for people? Companies typically believe that the BOTTOM LINE is the bottom line. In other words, they are convinced that the key to employee retention & satisfaction is MONEY. When someone resigns or threatens to leave, the

solution is usually to offer more money. As a professional recruiter, I have found few people making job changes for strictly financial reasons. Most often, money is not even the primary concern. Good managers usually recognize that money may not be the #1 issue, (because often it is not their own prime motivator). However, they often make the same type mistake that the company has made by assuming that what motivates them will also motivate their staff. Too often, employees become unhappy and end up changing jobs because a manager has misunderstood what that employee wants and needs for job satisfaction when what they leave seeking could have been found in their present job situation.

As a professional recruiter, I talk with people every day about career changes. I have the opportunity to listen to people talk about their priorities in life and career. Those things formulate their criteria for evaluating their present job and every other career opportunity presented to them. Most often, their concerns will fall into one of 7 categories. I will often have them rate those 7 in order of importance to help me know how I can best assist them. To illustrate the differences in individual motivators, rate the following in order of its importance to you starting with 1 as most important and continuing through 7 (least important):

Money_____
Benefits_____
Challenge_____
Security_____
Location_____
Potential for Advancement_____
Recognition_____
(Review results)

The results of this exercise demonstrates the problem. Just as we have different priorities, so will those we supervise. The key to effective management is understanding those differences and managing accordingly. For centuries, there has been a realization of the differences in behavioral styles. There have been numerous attempts to understand these different patterns and explain them.

Ancient astrology attempted to explain the differences by the alignment of the heavens at the time of a person's birth. They came up with 12 signs in 4 groups -earth, air, fire, and water.

Later, Hippocrates theorized the existence of 4 basic temperaments - phlegmatic, choleric, melancholy, & sanguine - determined by 4 factors - blood, phlegm, black bile, and yellow bile.

In 1923, Dr Carl Jung, Psychological Types, determined that people fit into one of 4 personality types -Intuit, thinker, sensor, or feeler.

Theories abound, but there seems to be one common thread: **Behavior styles can be grouped into 1 of 4 major categories.** There is a danger in becoming overly analytical. However, an understanding of the four basic behavioral styles can be a simple tool to help us evaluate where we and our staff are and how best to manage our people resources. After we've looked at the four styles, we're going to get back to our buddy Bart and his famous friends.

First style we will encounter is the **DIRECTOR:** He's a natural leader, assertive, responsible, self sufficient, & self motivated. The Director takes charge. He loves a challenge and is quick to analyze a problem and chart a course of action. RESULTS are what count most. He has no problem doing whatever it takes to get the job done, even if it means hurting people in the process. Directors can be insensitive, blunt, and often critical because they value tasks above relationships. They can be overbearing and uncompromising, but they are never unproductive. They can't stand wasted time and move at a fast pace. A Director demands a lot of himself and expects the same from those around him. Directors don't get ulcers, they give them. In the words of Toyota - WE ARE DRIVEN!

Very different from the Director is the **SOCIALIZER:** He is outgoing, spontaneous, animated, and playful. The Socializer loves people and hates ruts. He is talkative, stimulating, and full of fun. He likes to move at a fast pace, juggling lots of things at the same time. He is impatient and easily bored. Because he is adept at thinking on his feet and good at jumping to conclusions (amazingly, right ones more often than not), he is often disorganized and unprepared, relying on wit and charm to pull him through. He establishes rapport quickly and values relationships more than tasks. He is creative, impulsive, unpredictable, and easily bored. The Socializer can be emotional, yet often lacks seriousness.

The third behavioral type could be known as the **THINKER:** He is everything the Socializer is not. He is cautious and precise, conservative and consistent. He values facts and logic and takes a systematic, detail-oriented approach to tasks. The Thinker does not like to be rushed. He makes decisions only after exhaustive consideration of all the available information (even then is never sure he wasn't a bit hasty). He is non-aggressive and typically loyal (once he makes the initial commitment). Since he values tasks above relationships, he usually lacks people skills and is not comfortable in front of groups. His passion for detail can make him picky and at times a bit self-righteous. He is diligent and dependable when given a task and will do a complete and thorough job.

The last of the four personalities is the **RELATER:** He is warm, reliable, amiable, and pleasant. Relationships and people are his first priority. Relaters hate conflict and friction. They are loyal and cooperative, preferring the status quo to change. They are great listeners and will be supportive in listening

even if they don't agree, preferring to give in rather than to fight. They are non-assertive, patient, and often indecisive. Since acceptance and security are most important to the Relater, "**RISK**" is the consummate 4-letter word. He requires constant reassurance. **AVAILABLE** is his middle name, and he will have difficulty saying "no" to anyone, even someone he doesn't like.

Though we may at times adopt a different approach to certain social or business settings, each of us has a "core style" that we most naturally exhibit. That style can be easily identified by evaluating 2 factors:

1.)

OPEN

1
2
3
4

SELF CONTAINED

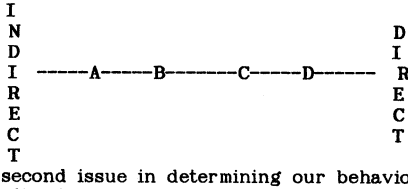
Each of us will find that we are naturally more comfortable being open with others about ourselves and our feelings, or that we will be somewhat closed or self-contained.

An **OPEN** person can be identified by: Frequent eye contact, spontaneous use of hands and gestures, animated facial expressions, free use of arms, legs, and body positions in communication...The Open person can wear his heart on his shirt sleeve. He shares his personal life and even his financial difficulties. He is intuitive, making decisions on the basis of his "gut" or his emotions. He is flexible and places people above tasks.

On the other hand, the **SELF CONTAINED** person is not quick to show his emotions. He is cautious in relationship building. He is work-oriented, well organized, and usually enjoys the planning process. He likes structure, guidelines, and procedures. He is typically more punctual and disciplined. Decisions are made on the basis of logic and rational thinking.

On a scale of 1 to 4, assess yourself according to your "openness quotient."

2)

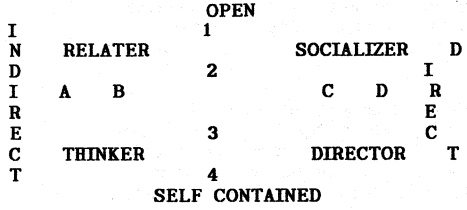


The second issue in determining our behavioral style relates to directness. A **DIRECT** person will typically be fast-paced and assertive. He is competitive and out-spoken. He is talkative and speaks fast with emphasis. He likes to control

and is dominant. He sees in black & white. He's impatient and likes to jump right into things.

The **INDIRECT** individual in contracts sees things in shades of gray. He is quiet and reserved. He is generally tactful, cooperative, and at times compliant. He avoids risk and seeks security. He is cautious, supportive, and generally a better listener than his counterpart. He is less competitive and does not like confrontation.

Using these 2 scales, we can find our basic style by incorporating the 2 factors into the chart below:



Using that same criteria it is easy for us to evaluate the people we work with to determine where they fit. If we take that information back to our cartoon characters who are less fictional than we care to admit, we can reach some amazing conclusions. Let's use what we've learned with Bart and his buddies.

1) **BART SIMPSON** - At fist glance we see an underachiever, proud, rebellious, a troublemaker, mischievous, creative, energetic, great at analyzing situations (problem is he uses that information to make things worse), Open: he'll say or do anything, Expressive; Direct: he loves confrontation & likes to bring attention to himself...

Where does he fit? Bart's **A SOCIALIZER.**

2) **HOMER SIMPSON** - Also an underachiever, but for different reasons, good intentions but poor results. Loyal, dependable, too proud to admit he needs help. He suffers from self recrimination. In analyzing the situation, he sees himself as a failure. "Openness quotient": more self contained (Remember the Christmas episode? No bonus. Rather than tell anyone, he goes out and gets 2nd job as store Santa, then resorts to gambling at the dog track.); "Directness quotient" Compliant to a fault. He takes it from the boss, his son, and everyone else in his life. Afraid to risk honesty, he avoids confrontation (Bart runs all over him because he won't stand up for himself, even to his own son)

Where does Homer fit? He's a **THINKER** (shock!!!)

3) **MARGE SIMPSON** - Caring, patient, loves routine, hates conflict, unwilling to face reality, a people pleaser, ("you will be popular as long you keep smiling"). She will do anything for anybody. "Openness quotient": open & honest to a fault. She wears her heart on her shirt sleeve, values people above tasks, & is emotional; "Directness quotient": She's compliant & avoids risk. She's supportive, a good listener and empathizer...

Who is Marge - **the RELATER.**

4) **LISA SIMPSON** - more of a loner. She sees people for the most part as an unnecessary evil. She's bright, creative, & not afraid to try anything; quick to criticize, sharp and cutting at times, talented, & restless. "Openness quotient": definitely more Self Contained; "Directness quotient" -likes to dominate or control, impatient, out spoken, sees in black & white- definitely more direct than indirect...

Who is Lisa - **the DIRECTOR.**

Knowing a person's style will not change their level of competence or motivation toward mutual objectives, but it can help us tailor our management style to maximizing our staff's potential. Each personality will require of us an adjustment in our approach based on our understanding of their "comfort zones" and "hot buttons." Our presentations & explanations, our task assignments, our corrective procedures will need to reflect that understanding.

BEHAVIORAL SUMMARY

CATEGORY	RELATER	SOCIALIZER	THINKER	DIRECTOR
BEHAVIOR	indirect & open	direct & open	indirect & self-contained	direct & self-contained
PACE	slow & easy	fast	slow & easy	fast
PRIORITY	relationships	relationships	task	task
FOCUS	building trust	dynamics of relationship	details & processes	results

DURRE PLACEMENT SERVICES

IRRITATION	pushy behavior	boring tasks	surprises	wasting time
FOR DECISION GIVE	guarantees & reassurance	incentives & testimonials	facts & details	options & outcomes
THEY QUESTION	effect on personal situation	who else uses it	how it works	what it does & by when
SPECIALTY	support	socializing	processes & systems	being in control
FOR SECURITY RELY ON	close relationships	flexibility	preparation	being in control
ACCEPTANCE DEPENDS ON	loyalty	playfulness	being correct	leadership skills
TO INCREASE FLEXIBILITY NEED TO	take risks!	control time & emotions	appreciate others	show concern for others

TWO TIMELY TRUTHS that will help you avoid some frustration in dealing with your own "Simpsons":

1) **WHAT YOU SEE IS WHAT YOU GET.** Education will not make a Bart into a Lisa or a Homer into a Marge. The best we can do is to make Bart a better Bart or to help Homer become a better Homer. It is far wiser to work within the framework of the personality than to try to force it to change.

2) **APPARENT WEAKNESSES CAN ALSO BE STRENGTHS.** It is too easy to focus on the negatives and miss the positives. A wise manager will discover what his people can do best, then let them do it. As the wise management consultant of "Jungle Book" said, Accentuate the positive. If you can't totally eliminate the negative; at least don't illuminate it publicly. Realize your staff's abilities & inabilities. **BART** may be a brat, but he has more creativeness in his little finger than Homer has in his entire body. If it is every channeled into a positive direction, he can be awesome!

And there's a lot to be said for **HOMER.** Teach him what to do and why, and it will be done...for the next 40 years or so. Repetitive tasks boring to Bart are fine for Homer. Be thankful for your Homer's. They may require more training but need less "maintenance" than your Barts or Lisas.

MARGE may not be witty or creative, but shes the oil that can keep the friction between a Bart and a Lisa from bursting into flame. She's got a jello mold for every occasion.

LISA may be obnoxious and lacking in people skills, but when she decides to do something, try and stop her. She is driven to succeed and won't be sidetracked. Whether its learning to play the sax or making the grade, she's up to the task.

The same will be true for the "Simpsons" in your shop.

If you have a "**BART**" (a **SOCIALIZER**), understand that he thrives on activity. He can't stand being bored. He loves to be challenged. He hates repetition. When possible, take advantage of his energy and his creativity. If he must be in a repetitive, maintenance role, sell it to him on the basis of "needing" his unique talents in that role. Give positive "testimonial" feedback of his value to you and his user base. He will thrive under pressure; he will be at his peak when juggling more than one responsibility. Recognize his quick analytical ability and utilize it. But make him accountable for time management and careful preparation. He is a competitor, so funnel his competitive energies toward dept. goals and objectives and reward him for achievement. (Rewards should include public recognition as much as monetary compensation). The Barts are the potential "stars" of the department but can leave it scarred instead if those creative energies cannot be harnessed.

The **HOMERs (THINKERs)** in your shop can be your biggest assets. They thrive on perfection and are perfunctory in their performance. They will better handle maintenance roles than any other personality group. They are great at researching projects and can usually find out everything you need to know (and more) to make a careful decision. However, recognize that if the decision is theirs to make, you may be in serious trouble. Since they are more cautious by nature and love processes and details more than decisions, projects left to thinkers may never get completed UNLESS you give them specific parameters, both objectives and time constraints. Understand if you are a Director, that the Thinker may have more questions and concerns than you care to deal with, but what may appear to be a waste of time now may actually save you countless more hours later in retraining or correcting. If you are dealing with a **MARGE, (RELATER)**, realize that she requires regular maintenance. Regular contact is critical since guarantees and reassurances are so important for her. And since the relationship with you as manager is usually more important than the task you have assigned her, she may be reluctant to share insights if you do not give her "permission" to tell you the truth. Relaters are good at sharing ideas for building and maintaining the shop "morale." Relaters are good at assessing how the user community will react to new systems. They can be valuable in manning "help desks" or training users, and during times of change, the Relater can be a good source in assessing "emotional fallout" brought along with those changes. However a word of caution is necessary: Although the Relater appears to be a good listener, his assessments may not be accurate as to the severity of the concerns. Remember, of the 4 types, he least likes RISK and most prefers the status quo so he can at times make mountains out of emotional molehills. Rest assured, the Relator will be loyal, prompt, in compliance with any and all rules, and will give you the best he has to offer. (Which

makes it worth the time to keep up with the maintenance.)

Your LISAs, (DIRECTORs), will prove invaluable for those times when the deadlines are close and the work load is great. Pressure is not a problem, and they love making decisions. Give them that tough assignment and they will find the fastest, most efficient way to get it done. They thrive on results, and they need little or no personal maintenance! If you need someone to sit in the closet and crank out code, look for a "Lisa". However, keep an eye out for emotional fallout that may accompany their efforts. User skills will not be their forte. Neither will they be overly concerned with the feelings of their co-workers. Since socialization is often considered a waste of time, don't expect them to be "gung-ho" about office parties and the like. Just keep them busy and challenged and they're happy.

SO IF I HAVE TO HIRE ONE OF THEM, WHICH ONE SHOULD I HIRE?

That answer depends on several factors: What will the person be doing? What kind of PACE must they keep? How much involvement will the person have in decision making? How important is it that they be able to work with users? What kind of user base will they be working with? What are the personality styles of your shop? What are your strengths? Your deficiencies?

A carefully planned job description, including personality and "people skills" requirements will help you make that decision.

Whoever you hire, make a commitment that you are going to get to know and understand them - to know what they value, where they are headed in their personal and professional lives -and then commit yourself as a manager to manage them accordingly. "Roll 'em, rawhide!" may work fine for driving cattle, but try to manage people that way and you will find yourself knee deep in smelly stuff.

However, I am convinces that real managers who take into account the differences of the individuals they supervise and manage them accordingly, will not only get their respect but peak performance as well. Zig Ziglar has it right when he says, "You can get everything you want in life if you help enough other people get what they want." Especially if those "people" are the ones you supervise every day. Then you will be able to exclaim along with the great Bartman himself,

"COWABUNGA, MAN!"

Working With Difficult People

Louis R. Mills, CDP

Some people are always difficult to deal with. And these people account for most of the stress in your life. Why are they difficult? Are they stupid? Mean? Arrogant? Do you find yourself avoiding them or trying to work around them? Doesn't that make your job harder?

People are "difficult" when they make us react emotionally, usually with anger and frustration. How can you minimize or eliminate conflict when dealing with difficult people? Should **you** be difficult in return? What triggers their behavior? How do you keep your cool when everyone else is screaming?

Knowing what makes some people "difficult" is the first key. Learn which behaviors aggravate you and how to change your response. Gain confidence in dealing with confrontations and arguments. Apply a little healthy psychology to change your reaction to difficult people and watch them change, too.

INTEREX HP Users Conference 1991
San Diego, California
Paper Number 6102

When Is A RUG Not Something You Walk On Or Beat?

David L. Largent

The N.G. Gilbert Corporation
700 South Council
P.O. Box 1032
Muncie, Indiana 47308-1032
U.S.A.

Telephone 317/284-4461 Facsimile 317/288-2125
Telex 361285

1 *Introduction -- "Is this for me?"*

This paper will venture into the nontechnical world, yet provide you a way of obtaining technical information. We will talk about people, yet tell you how to find out about bits and bytes. Our discussion will not apply to any particular revision level of the operating system, yet will provide you a way to learn about the most recent changes. In short, we will talk about RUGs -- Not the kind you walk on or beat, but rather the kind you become an integral part of.

On the following pages will be found answers to questions related to Regional Users Groups (RUGs) such as:

- What are they?
- How can they help me?
- How can I help them?
- How are they organized?
- Are they worth investing in?

These questions will be answered by comparing and contrasting a number of different RUGs to find out what they are, and how we can help each other grow. I will be using the acronym RUG in this paper to refer to both business and technical users groups. Historically, the business groups have been referred to as RUGs -- regional users groups, and technical groups have been referred to as TUGs -- technical users groups. My choice of RUG is not meant to offend the "techies"; it simply made for a better paper title.

This paper is directed at individuals who are not currently participating in a RUG. Those individuals may be new users, or they may be people with ten years of experience! Even current RUG members are encouraged to read this paper, for they may find it interesting to learn how other groups operate. I also welcome any comments from individuals who either agree, or disagree, with what I present here. It is this exchange of information and opinions which permits us all to grow. So take some time to find out why you should invest in a RUG and make your path a little softer to walk on.

1.1 Where did the data for this paper come from?

Although I have had exposure to a large number of users groups because of my involvement on the INTEREX Affiliates Council Steering Committee (ACSC), I still felt a need to get information directly from as many groups as possible. One thing I have learned during my years on the ACSC is that there is not a "typical" RUG. Every group does something a little different from the next. To gather information, I first formulated a questionnaire that was sent to all North American INTEREX-affiliated groups. I supplemented the questionnaire with telephone calls to all affiliated groups that did not respond to the questionnaire, and to all non-affiliated groups in North America. (Yes!, it took a good deal of time, but I feel the end result has been worth the effort.)

Presented in this paper is the compilation and distillation of responses from nearly fifty different groups. It represents a good cross section of users groups in North America. Numerous large/small, business/technical, affiliated/non-affiliated, and Canada/U.S. groups are represented. While the paper presents general information about all groups, an appendix provides specific information about the groups I was aware existed on May 1, 1991.

2 RUGs -- What are they?

Beyond the "obvious" of a RUG being a group of people who get together periodically, what is a RUG? To describe what a RUG is, we will approach the question from three vantage points:

- What is the purpose of a RUG?
- What is the value of a RUG to an individual?
- What is the value of a RUG to an individual's organization?

2.1 *The purpose of a RUG.*

One way of describing a RUG is to state its purpose. The responses I received fell into three categories:

- Information exchange/education
- Communication/networking
- Voice to Hewlett-Packard Company (HP)

2.1.1 *Information exchange/education.*

Perhaps the most commonly mentioned purpose of a RUG was to share information amongst the members. The information might be technical in nature, such as recent operating system changes, or it might be more general, such as talking about managing stress in the Data Processing shop. A RUG also provides for the sharing of ideas.

Nick Demos of BWRUG provided the following purpose for a RUG:

> Encourage the development of, and provide a medium for exchange of, techniques, applications software, documentation, and procedures for HP computers.

A booklet published by INTEREX titled *Starting a Regional Business/Technical Users Group Guide* provides this description:

> [A RUG] creates a forum where not only technical, but business and management information can be exchanged.

The bottom line is that a RUG is a great place to exchange information and be educated.

2.1.2 *Communication/networking.*

Beyond the obvious purpose of information exchange and education, the idea of promoting communication between members consistently came up. This might take the form of sharing experiences (good or bad) with one another, or simply getting users and managers to talk to each other. A RUG often provides a forum for meeting other users in the area and encouraging cooperation and mutual help amongst users of HP computers.

2.1.3 *Voice to Hewlett-Packard Company.*

RUGs always have the capability to provide a coordinated voice to HP to address specific concerns of either an immediate or long-term nature. This coordinated voice can be directed at the local level, as well as at the corporate level.

2.2 *The value of a RUG to an individual.*

Another way of describing a RUG is to state the value it has to an individual -- that is, Why do people participate in a RUG? The responses I received fell into three categories:

- Information exchange/education
- Communication/networking
- Personal reasons

I find it interesting that the responses given to this question so closely parallel the responses provided for the purpose of a RUG.

2.2.1 *Information exchange/education.*

The responses here are extremely similar to those given earlier for the purpose of a RUG. Interesting speakers provide a way to pick up new information, and to learn of new software and/or hardware. They also provide a forum for learning of problems others have experienced, and the solutions to those problems. By talking with others, you will likely find that your problems are not as unique as you thought they were.

2.2.2 *Communication/networking*

Again, the responses are very similar to the purpose statement. Being able to meet with other users and/or managers, and having contact with experienced users were both mentioned a number of times as I talked with people. It was also observed that it is sometimes quicker and/or easier to call a local user rather than a phone-in consulting service (PICS) when you have a question or problem. The bottom line here is that you are able to establish a network of contacts to help you through your day-to-day challenges.

2.2.3 *Personal reasons.*

A motivating factor for participating in a RUG is often more personal or selfish in nature than anything else! Being human beings means that there is some degree of selfishness in each of us. A RUG can offer peer recognition of your individual achievements and involvement. We may get a sense of worth because we are able to give something to others.

Often times, a RUG offers opportunities for career advancement through associations made in the group. A small, but growing, number of groups are offering academic scholarships. As a RUG member, you may be eligible to apply for one. Lasting friendships within the group are often established. And on the purely selfish side, RUGs provide a way to get out of the office and have some "good times".

2.3 *The value of a RUG to an individual's organization.*

Moving our vantage point a little bit, permits us to take a look at RUGs from the organization's point of view. The responses again fall into familiar categories, with a new one thrown in:

- Information exchange/education
- Voice to Hewlett-Packard Company
- Business reasons

2.3.1 *Information exchange/education.*

Ditto of the above corresponding sections! RUGs provide access to new ideas to address business problems, and provide access to users with solutions. A promotional brochure used by INRUG states:

> No matter what questions arise, or what problems you may face, chances are some member has encountered the same thing and will be more than happy to share his or her experiences with you. Imagine how valuable it would be to know some experienced local HP users to call when you have a question or problem.

2.3.2 *Voice to Hewlett-Packard Company*

In a January, 1991 article in *INTEREXPress*, Diane Amos covered this section as well as I could -- so why re-invent the wheel? She stated:

> The united voice gives HP a clearer view of user priorities for hardware, software and maintenance. HP will sit up and listen when confronted with a serious problem that has plagued several shops. They may not have been aware of the problem, especially if it has been addressed by one or two users. However, when encountered by strength of a RUG, HP takes it seriously.
>
> Often the HP managers are hearing about a problem for the first time and the users are able to speak directly to the manager who can solve it for them. This kind of interchange is a win-win situation for both HP and the HP user.
>
> HP managers can hear from the "horse's mouth" what's going on in the DP shops, and not just what is reported to them.
>
> In an informal setting, RUGs offer Hewlett-Packard the opportunity to demonstrate its latest equipment and software firsthand.

All these comments can be applied equally well to any other vendor in the HP world. If a vendor is involved with a RUG in a positive way, everyone will come out to the good.

2.3.3 *Business reasons*

Generally, when it comes to justifying an organization's involvement with a RUG, selfish reasons pop up again: What's in it for the organization? A number of reasons surfaced as I talked with groups. The most common response: They get a better educated, more well rounded employee who provides better performance because of their contacts and knowledge. Often times, information about individuals looking for a job will be available. In an article in the March, 1991 *IMRUG Newsletter*, Christopher Hart stated:

> When it comes to return on investment, a users group meeting is probably one of the higher returns on investment a company can possibly see. The fees charged for the meeting are to cover costs only... Beyond the technical presentations, adequate time is allowed for networking and getting to know other users within the users group. The amount of knowledge that is shared at a typical meeting goes far beyond the technical presentations.

One last avenue to explore is business leads. If your organization sells an appropriate product, or provides an appropriate service, you may well benefit from simply having a presence at RUG meetings. Diane Amos states (in the *INTEREXPress* article):

> The obvious benefit [for a vendor] is solidifying the customer relationship face-to-face [at a RUG meeting or conference] when the only previous contact may have been on the telephone.

3 *How are RUGs organized?*

Up to this point we have been subjectively exploring what a RUG is, and why individuals and organizations become involved with them. Let's now turn to some concrete, "dyed-in-the-wool" type information about RUGs.

3.1 *Governing organization.*

Every RUG has one or more individuals that are responsible for its ongoing success. For most groups, this is an elected Board of Directors. Also very common is either an elected or volunteer committee. Although I did not get any of them to admit it, there are a few groups that appear to be run by a dictator.

3.2 *RUG membership.*

Virtually all RUGs provide for an individual membership. Many also have a corporate membership, where a number of individuals from the same organization receive member services. A few have only a corporate membership, while still others have chosen to establish membership based on whether you are an INTEREX member or not. Some RUGs require you to be a member of INTEREX before you may join the local group.

The cost for annual individual membership in a RUG ranges from free to $35.00, with the arithmetic mean being $20.00 for those groups that charge. Corporate membership ranges from free to $150.00, with the arithmetic mean being $60.00 for those that charge. Over sixty percent of the groups I spoke with do not charge for membership.

3.3 Number of members.

The size of RUGs varies greatly. The smallest group I am aware of has 15 members, while the largest has over 1,500! The arithmetic mean is somewhere around 200 people. Over forty percent of the groups I spoke with have membership counts of 100 or fewer people.

3.4 Funding and support of the RUG.

Finances are often a major concern for RUGs, just like they are for most of us in our daily lives. So where do RUGs get their income? One common source we already mentioned is from annual membership dues. Another common source is fees collected from meeting attendees. If the RUG publishes a newsletter or meeting announcement, income is likely available from advertising placed in the publication. RUGs also often provide "blind mailing" services to advertisers. If the RUG is affiliated with INTEREX, there is a rebate available to the RUG for each INTEREX member in the RUG's area. The current rate is $5 per year for each INTEREX general membership, and $25 per year for each INTEREX site membership.

3.4.1 Support from Hewlett-Packard Company (and other vendors).

While a number of vendors support many of the RUGs in varying degrees, HP is mentioned most often when someone talks about a vendor supporting their group in some way. This support may take the form of actually funding some or all of the expenses of the RUG, or may simply be a presence of HP employees at RUG meetings. Over forty percent of the groups I spoke with hold one or more of their RUG meetings at their local HP office. Very often HP will provide the snacks for breaks during the meeting. Door prizes are also commonly provided by HP.

Other support services HP provides to many of the RUGs include furnishing speakers for the RUG meetings or conferences and providing access to HP computing equipment for demos at meetings and conferences. Many groups would have a difficult time surviving if it weren't for the support that Hewlett-Packard Company or some other vendor provides them.

4 How can we help each other grow?

If you really get involved in a RUG, it becomes a growing, win-win situation for all involved. If you occasionally show up at meetings and sit there like a bump on a log, it will likely become a lose-lose situation. You will not get something for nothing -- you have put a little effort into it.

4.1 How can a RUG help me?

The answer to that question comes from:

- Attending RUG meetings and conferences
- Reading the RUG newsletter and other mailings
- Getting involved in Special Interest Group activities

4.1.1 Attend RUG meetings and conferences.

As I have alluded to earlier, one of the best ways to utilize the benefits that are provided by a RUG is to attend RUG meetings and conferences on a regular basis. In this way, you can start to share in the information exchange and education that is available.

4.1.1.1 Meeting types.

I found a wide diversity of meeting types as I talked with the different groups. RUG gatherings ranged from two-hour meetings to three-day conferences. The most common meeting formats are a half-day meeting and full-day meeting. Next in line are dinner/evening meetings followed next by one-day conferences. Meetings typically consist of one or two technical presentations, while conferences typically have many more technical presentations, and a vendor show as part of the event.

4.1.1.2 Meeting schedules.

Just as there is a wide diversity in meeting types, there is also a large diversity in how many meetings a RUG holds during a year. The number ranges from one to twelve. Nearly half of the RUGs I spoke with meet quarterly; another one-fifth hold meetings every two months. Eleven percent of the groups hold two meetings a year. Fourteen percent meet either three or five times a year, and the remaining groups meet one, eight, ten, or twelve times during a year.

4.1.1.3 Meeting topics.

Meeting topics are all over the map as well. Topics like system management, systems analysis, PC integration, networks, system performance, CASE tools, writing skills, SQL/relational database, data communication, object oriented programming, UNIX, bar coding, HP NewWave, desktop publishing, and disaster recovery is but a sampling of the presentations at recent RUG meetings.

4.1.1.4 Meeting highlights.

When I asked about highlights of past meetings, I was showered with glowing success stories. One of the most recent success stories occurred in the fall of 1990: the All-Texas Conference. All of the RUGs in Texas got together and held one large, multi-day conference in Galveston, Texas. It is expected to become an annual event; Austin is the venue for 1991. Another highlight mentioned by a number of RUGs was a very productive HP management roundtable discussion. Other groups reported good success at including both professional and technical presentations in their meetings. Fourth generation language "shootouts" have been very popular presentations in the not-too-distant past.

4.1.1.5 Cost to attend a meeting.

The cost to attend a meeting or conference varies widely from group to group. To attend a half- or full-day meeting, the cost ranges from free to $70.00, with the arithmetic mean being $20.00 for those groups who charge for meeting attendance. The fee for attending a conference ranges from $15.00 for a one-day conference to $300.00 for a three-day conference. It is very common for RUG members to be charged a discounted price for the meeting. I have heard a number of times that users group meetings and conferences are the most cost-effective educational opportunity available.

4.1.1.6 Typical meeting attendance.

Just as the sizes of RUGs vary, so do the typical attendance figures. The lowest typical attendance for a meeting I am aware of is ten people, and the highest is 200, with the arithmetic mean near 50 people. For conferences the figures are 60, 500, and 175 respectively.

4.1.2 Read the RUG newsletter and other mailings.

What better way is there to keep informed of what your RUG is doing than to read the groups's publications?! These typically contain technical articles, product announcements, problems to watch for, answers to your questions, and even the occasional joke or amusing story. Seventy percent of the groups I talked with publish a newsletter of some kind. For some RUGs this is a one page document; The other end of the scale includes newsletters that are 35 pages in length. The arithmetic mean is fourteen pages. Of the groups that publish a newsletter, nearly two-fifths of the newsletters contain ten or fewer pages per issue.

4.1.3 Get involved in SIG activities.

Many of the RUGs have mini-users groups within the users group. I am referring to SIGs: special interest groups. The SIGs make up a subset of the RUG that has a common interest in some topic or product. The activity of SIGs varies from almost nonexistent to extremely active. I am aware of SIGs that exist for the following topics

or products in one or more of the RUGs: 4GL, MPE, Powerhouse, PROTOS, Speedware, System Management, Transact, and UNIX.

4.2 How can I help the RUG?

The biggest help you can be to the RUG is to be involved. When volunteers are asked for, step forward. Ask those "dumb questions" in the meetings, so that all of you will get an answer. Run for an elected office. Encourage nonmembers to join the RUG. Write articles for the newsletter. Host a meeting at your organization's site. Offer to help in organizing the next meeting. Offer to make a presentation at the next meeting! Most people find the more they put into a RUG, the more they get out.

5 Conclusion -- "Should I invest in a RUG that I can't walk on or beat?"

Yes! Get involved. If there is not a RUG in your area, start one! You have so little to lose, and so much to gain. Diane Amos states (in the *INTEREXPress* article):

> Participating in a RUG provides both the HP user and the vendor a means to keep informed in the HP world. The user can read through press releases to see what's going on, but hearing it firsthand is more useful. Attending a RUG meeting can be as good as formal training. The vendor has the opportunity to learn what the competition is up to. It also gives HP employees a chance to talk with several of their customers all in the same place.

Christopher Hart summed up my feelings when he wrote (in the *IMRUG Newsletter*):

> Take time to support the users group by your attendance, contributions to the newsletter, etc.. With a strong users group, everyone gains.

Amen!

Acknowledgements

The following people were contacted as I gathered information about the RUGs in North America. I very much appreciate the time each of them spent with me. Without their willingness to share information, this paper would not have been possible.

Tahmina Ahmed - SCRUG
Brian Becker - DeltaRUG
René Chrétien - QCRUG
Randy Cliff - BCRUG
Dave Crutchley - SIERRARUG
Nick Demos - BWRUG
John Evans - WNYRUG
Donna Fairchild - LMRUG
Terry Floyd - CENTEXRUG
Connie Grenz - ORERUG
Christopher Hart - IMRUG
Jerry Head - GNYRUG
Michel Kabay - MTLRUG
Wanda Komorowski - TUG
Manny Mankowski - NOWRUG
David Moirao - BARUG
Wendell Payne - DARUG
Mike Shelton - NTRUG
Gary Sogar - CINMUG
Dave Tewel - SPORUG
Terry Warns - SMUG
Jim Weiss - NEORUG
Charles Weston - MARUG
Jane Zubovich - CORUG

Flo Barley - CILRUG
Isaac Blake - AZRUG
Steve Click - DWFTUG
Nancy Coryell - OKRUG
Chris Dale - RMRUG
Kim Doolittle - AARUG
Bud Evers - ARUG
Zachary Fleck - SLRUG
Jerry Fochtman - GHRUG
Dale Halterman - SCRUG
Doug Hawkland III - WBTUG
John Hurt - STRUG
Ronald Keaton - PITTRUG
Anne Long - IARUG
Gabrielle Markvorsen - OVRUG
Maurice Meilleur - CBRUG
Linda Roatch - MINNRUG
John Smedinghoff - NMRUG
JC Strote - SACRUG
Leslie Virgilio - USNYRUG
Lee Webb - BRUG
John Werner - NECRUG
Jackie Williams - FLORUG

Appendix: North American Hewlett-Packard Users Groups

(Group Name, Area/Equipment Served, Newsletter, & Scholarship Information)

Short Group Name	Full Group Name	Geographic Area Served	Equipment Served	Affiliated with INTEREX (ITX) (1)	Year Formed	Form of Government	Newsletter (Issues per Year)	Pages per Newsletter Issue	Scholarship (Amount Awarded) (2)
AARUG	Adirondack Area Regional Users Group	Northeastern New York	3000	Yes	1981	Committee	None	None	
APG TUG	Aberdeen Proving Grounds Technical Users Group	Maryland	1000	No					
ARUG	Alabama Regional Users Group	Alabama	3000	Yes	Last half of 1970s	Board	4	8	None
AZRUG	Arizona Regional Users Group	Arizona	3000	Yes	1980	Board	6	8-10	
BARUG	Bay Area Regional Users Group	San Francisco, California bay area	3000	Yes	1974	Board	None	None	
BCRUG	British Columbia Regional Users Group	British Columbia, Canada	3000	Yes	1984	Executive Board	4	25-35	None
BRUG	Border Regional Users Group	Southeastern New Mexico, Western Texas, Northeastern Mexico	1000, 3000, 9000, PC	No	1989	Committee	None	None	None
BWRUG	Baltimore/Washington Regional Hewlett-Packard Users Group	District of Columbia, Maryland	3000	Yes	1977	Board	4	24	Every year: $5,000 spread over 4 years
CBRUG	Columbia Basin Regional Users Group	Eastern Washington	3000	No	1982	A Coordinator	4	1	None
CENTEX-RUG	Central Texas Hewlett-Packard Regional Users Group	Central Texas	3000	Yes	1989 or prior	Board (One member must use Unix)	4	10-20	
CILRUG	Central Illinois Regional Users Group	Central Illinois	3000	Yes	1983	Board + Two User Advisors	None	None	None
CINMUG	Cincinnati Metropolitan Users Group	Cincinnati, Ohio area	3000	Yes	1985	Executive Committee	4	6	None
CORUG	Central Ohio Regional Users Group	Central Ohio	3000	Yes	1981	Board	4	16	None
CRUG	Chicago Regional Users Group	Northeastern Illinois (100 mile radius of Chicago)	3000	Yes	1980	Steering Committee	3	8	None

Appendix: North American Hewlett-Packard Users Groups
(Membership, Meeting & Other Information)

Short Group Name	Types of Membership	Annual Cost of Membership (2)	Number of Members	Meetings per Year	Type of Meeting (3)	Location of Meeting	Cost to Attend Meeting (2)	Typical Attendance	Other Comments
AARUG	Individual	Free	30	2	½-Day	HP Office	Free	10	
APG TUG				2-4					
ARUG	Individual, Corporate	$25 (ITX members free)	60	4	Full-Day	Member's Site, Hotel	Free	20-30	
AZRUG	Individual	Free	150+	6	½-Day	Various	Free	20-50	Each meeting includes both a professional & a technical topic.
BARUG	Individual	Free	1500	4	3 Full-Day & 3-Day Conference	Hotel & Conference Center	$30, $150	100, 500	Also part of the 2-day NEVCAL Conference. SIGs: MPE, 4GL
BCRUG	Corporate		175	4	3 Dinner & 1-Day Vendor Show	Conference Center	$35 ($CD)	40	
BRUG	Individual	Free	30	4	½-Day	HP Office	Free	15-30	
BWRUG	ITX Member, Non-member	Free, $25	250	6-8	2 ½-Day, 2 Full-Day, 3 Dinner, & 1-Day Conference		$25-$70	50-70, 50-70, 50-70, 150-175	SIGs: Unix, System Management.
CBRUG	Individual	Free	40	4	½-Day	Member's Site & Restaurant	Cost of meal	20	Meetings generally consist of round-table discussions. Each meeting is hosted by a member company.
CENTEX-RUG	Individual	Free	100	3	Full-Day	Hotel	$25	40	Meetings include 15 minute Vendor Presentations during the first hour. Also part of the 2-day All-Texas Conference.
CILRUG	Individual	Free	140	6	½-Day	HP Office, Member's Site	Free	10-20	
CINMUG	Individual, Company	$20, $50	10 Individuals, 90 Companies	5	Evening	Hotel	$15-$20	60-80	
CORUG	Individual, Corporate	$10, $25 (See comment)	100	4	½-Day	Hotel	$5	30	Annual membership for first ITX member from an organization is free.
CRUG	Individual	Free	400	4	3 Full-Day & 1-Day Conference	HP Office, Conference Center	Free	60-120, 250	Two scholarships awarded in 1990.

Appendix: North American Hewlett-Packard Users Groups
(Group Name, Area/Equipment Served, Newsletter, & Scholarship Information)

Short Group Name	Full Group Name	Geographic Area Served	Equipment Served	Affiliated with INTEREX (ITX) (1)	Year Formed	Form of Government	Newsletter (Issues per Year)	Pages per Newsletter Issue	Scholarship (Amount Awarded) (2)
DARUG	Dayton Regional Users Group	Dayton, Ohio area	3000	No	1987	Planning Committee	6	5	None
DeltaRUG	Delta Hewlett-Packard Regional Users Group	Louisiana, Southern Mississippi	3000	Yes	1984 or prior	Board	4	20-24	$2,500
DFW TUG	Dallas/Fort Worth Technical Computer Users Group	Northern Texas	Apollo, 1000, 9000	Yes	1981 or prior	Officers	12	1	None
East Tennessee TUG	East Tennessee Technical Users Group	Eastern Tennessee	1000	No					
EIDAHO-TUG	Eastern Idaho Technical Users Group	Eastern Idaho	1000	No					
FLORUG	Florida Regional Users Group – Hewlett-Packard Computers	Florida	3000, 9000	Yes	1983 or prior	Board	4	16	None
GHRUG	Greater Houston Regional Users Group	Houston, Texas area	3000	Yes	1983	Board & Committee	4	28-32	$1,800
GNYRUG	Greater New York Regional Users Group	New Jersey, New York, New York area	3000	No	1977	Board	4	20	None
IARUG	Iowa Regional Users Group	Iowa	3000	Yes	1983	Board	5	25-30	
IMRUG	Intermountain Regional Users Group	Idaho, Utah, Wyoming	3000	Yes	1981	Board	4	8-12	
INRUG	Indiana Regional Users Group	Indiana	3000	Yes	1977	Board	6	22-26	$1,500
KYRUG	Kentucky Regional Users Group	Kentucky	3000	Yes	1983 or prior				
LMRUG	Lake Michigan Regional Users Group	Western Michigan	3000	Yes	1979	Board	5	6-12	None

Appendix: North American Hewlett-Packard Users Groups
(Membership, Meeting & Other Information)

Short Group Name	Types of Member-ship	Annual Cost of Member-ship (2)	Number of Members	Meetings per Year	Type of Meeting (3)	Location of Meeting	Cost to Attend Meeting (2)	Typical Attend-ance	Other Comments
DARUG	Company	$25 for first member, $10 for additional members from an orga-nization	55	5	Dinner	HP Office	$15, Non-member: $25	20-25	
DeltaRUG	Individual	Free (See comment)	35 Organi-zations, (125 Indi-viduals)	4	Full-Day	HP Office	$10	15-25	Must be ITX member to join. SIGs: Powerhouse, PROTOS, TRANSACT.
DFW TUG	Individual, Corporate	$12, $100 (for up to 10)	35	12	Evening	HP Office	Free	5-50	
East Ten-nessee TUG									
EIDAHO-TUG									
FLORUG	Individual	Free (See comment)	115	1	2-Day Conference	Hotel	$150	100	Must be ITX member to join. Local group meetings held throughout the year.
GHRUG	Individual	Free	400+	2	1-Day Conference	Hotel/ Conference Center	ITX Member: $35, Non-member: $45	80-100	Also part of the 2-day All-Texas Conference.
GNYRUG	Individual	Free	950	4	Full-Day	Hotel	$45	150-200	
IARUG	Individual	$25	60	5	Full-Day	HP Office, Member's Site, Amana Colonies	$15	35-40	
IMRUG	Individual	Free	325	4	Full-Day	Hotel	$30	80-140	
INRUG	Individual, Corporate	$20, $50 (ITX rebate for member applied against dues.)	115	6	½-Day	HP Office or Hotel	Free	40-70	Winner of 1989 & 1990 Golden Scroll Award for newsletter.
KYRUG									
LMRUG	Individual	Free	100	5	4 Evening & 1-Day Conference	Hotel	$15	45, 85	A very close network for sharing ideas/solving problems has devel-oped within the group.

Appendix: North American Hewlett-Packard Users Groups

(Group Name, Area/Equipment Served, Newsletter, & Scholarship Information)

Short Group Name	Full Group Name	Geographic Area Served	Equipment Served	Affiliated with INTEREX (ITX) (1)	Year Formed	Form of Government	Newsletter (Issues per Year)	Pages per Newsletter Issue	Scholarship (Amount Awarded) (2)
MARUG	Mid-Atlantic Regional Users Group	North Carolina, South Carolina, Virginia	3000, 9000	Yes	1981	Board	4	30	$2,000 given to a school with HP3000 to establish a scholarship
MIDRUG	Midwestern Regional Users Group	Southwestern Iowa, Eastern Kansas, Western Missouri, Southwestern Nebraska	3000	No	1989	Board	4	4-5	None
MINN-RUG	Minnesota Regional Users Group	Minnesota	3000	Yes	1982	Board	3-4	1-2	None
MTLRUG	Montreal Regional Computer Users Group	Montreal, Quebec, Canada	3000, 9000, PC	Yes	1979	Executive Committee	4	16-24	None
NECRUG	Northeast Central Regional Users Group	Delaware, New Jersey, Eastern Pennsylvania	3000, business 9000, PC	Yes	1977	Board	3	8-12	None
NEORUG	Northeast Ohio Regional Users Group	Northeastern Ohio, Northwest Pennsylvania	3000	No	1977	Board	4	8	None
NERUG	New England Regional Users Group	Central & Northern Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont	1000, 3000, 9000, PC	Yes	1984 or prior	Board	6	16	
NMRUG	New Mexico Regional Users Group	New Mexico	3000	No	Mid 1980s	Board	6	6	None
NOWRUG	Northwest Regional Users Group	Washington (West of the Cascades)	3000	Yes	1983 or prior	Board	4	2-4	None
NTRUG	North Texas Regional Users Group	Northern Texas	3000	Yes	1983 or prior	Board	4	20	None
OKRUG	Oklahoma Regional Users Group	Northeastern Oklahoma	3000	Yes	1981 or prior	Board	6	8	None
ORERUG	Oregon Regional Users Group	Oregon, Southwestern Washington	3000, 1000	Yes	1981	Board	6	16	None

Appendix: North American Hewlett-Packard Users Groups
(Membership, Meeting & Other Information)

Short Group Name	Types of Membership	Annual Cost of Membership (2)	Number of Members	Meetings per Year	Type of Meeting (3)	Location of Meeting	Cost to Attend Meeting (2)	Typical Attendance	Other Comments
MARUG	ITX Member, Non-member	Free, $25	1000+ (mailing list)	4	1½-Day Conference	Hotel	$80+	60-70	Meeting locations rotate through the 3 state area. Social/cultural events are tied in with meetings. Winner of 1990 Golden Scroll Award for newsletter.
MIDRUG	Individual, Corporate	Free	75	4	½-Day	HP Office	Free	45	Meetings held in conjunction with ½-day HP Installed Base Seminars to make a full-day meeting.
MINN-RUG	ITX Member, Non-member	Free, $25	125	3	2 ½-Day, & 1½-Day Conference	Hotel	Free, $95	30, 100	
MTLRUG	Individual, Corporate	$30, $60 + $15 per meeting attendee ($CD) (See comment)	250 + 100 Vendors	4	Full-Day	Hotel	$30-$60 (CD$)	60-90	The membership fee is a one-time cost – not annual. Meetings include 15 minute Vendor Presentations during the first hour. French-speaking group.
NECRUG	Individual, Corporate	$10, $25	1600	4	3 Full-Day or Dinner & 3-Day Conference	Hotel, Casino	$35-$50, $300	55-80, 250	SIGs: PC, Cognos, Manufacturing.
NEORUG	Individual, Corporate	$25, $75	100	4	½-Day	HP Office, Hotel	$5, non-member: $10	80-100	The summer meeting is held on a boat.
NERUG								20-40	
NMRUG	Individual, Corporate	$35, $150 (For up to 5 people)	50	6	Evening	HP Office, Restaurant	Free, Non-member: $15	30	
NOWRUG	Individual	Free	800 (mailing list)	3	2 Evening & 1-Day Conference	Hotel, Convention Center	$10, $65-$80	20-50, 150-200	
NTRUG	General, Corporate	Free (See comment)	120	3	2 Full-Day & 1-Day Conference	HP Office, Hotel	Free, $15-$35	70, 100	Must be ITX member to join. Also part of the 2-day All-Texas Conference.
OKRUG	Individual, Corporate	$10, $100, (ITX members free)	100	6	Evening	Restaurant	$15	40	
ORERUG	Individual	Free	200	2	2-Day Conference	Resort	$225	60-70	

Appendix: North American Hewlett-Packard Users Groups

(Group Name, Area/Equipment Served, Newsletter, & Scholarship Information)

Short Group Name	Full Group Name	Geographic Area Served	Equipment Served	Affiliated with INTEREX (ITX) (1)	Year Formed	Form of Government	Newsletter (Issues per Year)	Pages per Newsletter Issue	Scholarship (Amount Awarded) (2)
OVRUG	Ottawa Valley Hewlett-Packard Regional Users Group	Eastern Ontario, Western Quebec, Canada	1000, 3000, 9000, PC	Yes	1980	Executive	None	None	None
PITTRUG	Pittsburgh Regional Users Group	Western Pennsylvania	3000	Yes	1982	Board	4	4	None
Pittsburgh RTE TUG	Pittsburgh RTE Technical Users Group	Pittsburgh, Pennsylvania area	1000, Apollo	No					
QCRUG	Quebec City Regional Users Group	Quebec City, Quebec, Canada	3000	Yes	1986	Committee	None	None	
RMRUG	Rocky Mountain Regional Users Group	Colorado	3000	Yes	1978 or prior	Board	4	18	
SACRUG	Sacramento Regional Users Group	Sacramento, California area	3000	Yes	1984	Board	4	13-15	None
SALT-TUG	Salt Lake Technical Users Group	Utah	1000, 9000	No					
SCRUG	Southern California Regional Users Group	Southern California	3000	Yes	1978	Board	6	16	$1,000+
SERUG	Southeastern Regional Users Group	Georgia	3000	Yes	1985 or prior				
SIERRA-RUG	Sierra Regional Users Group	Northwestern Nevada, adjacent California	3000	Yes	1986	Officers	4	3	None
SLRUG	Saint Louis Regional Users Group	Saint Louis, Missouri area	3000	Yes	1983 or prior		4	16	Yes
SMUG	Southeastern Michigan Users Group	Southeastern Michigan	3000	Yes	Late 1970s	Officer Group	None	None	None
STRUG	South Texas Regional Users Group	Southern Texas	3000	No	1980 or prior	President	None	None	None
SPORUG	Spokane Regional Users Group	Northern Idaho, Inland Northwest Washington	3000, PC	Yes	1984 or prior	Officers Group	10	4-8	None
SVRUG	Sierra View (San Joaquin Valley?) Regional Users Group	Fresno, California area	3000, 9000, PC	Yes	1983 or prior	Executive Committee	None	None	None

When Is A RUG Not Something You Walk On Or Beat?

6102-18

Appendix: North American Hewlett-Packard Users Groups
(Membership, Meeting & Other Information)

Short Group Name	Types of Membership	Annual Cost of Membership (2)	Number of Members	Meetings per Year	Type of Meeting (3)	Location of Meeting	Cost to Attend Meeting (2)	Typical Attendance	Other Comments
OVRUG	Individual	Free	250-300	4	Full-Day	HP Office	Free	20-30	
PITTRUG	Individual	Free	130	4	½-Day	Member's site	Free	10	Each meeting is hosted by a member company.
Pittsburgh RTE TUG				4				20-30	
QCRUG	Users, HP Employees	$30 ($CD)	35	4	3 ½-Day & Full-Day		$20 ($CD)	25-30	French-speaking group.
RMRUG	Individual	Free	390	4	3 Full-Day & 1-Day Vendor Show		$15	50-100	SIG: Speedware.
SACRUG	Individual	Free (See comment)	75	4	½-Day & Full-Day		$5-$15	35	Must be ITX member to vote. Has great access to the HP Performance Lab. Also part of the 2-day NEVCAL Conference.
SALT-TUG				1-2				20-30	
SCRUG	Individual, Corporate	Free (See comment)	425	4	2 1- or 2-Day Seminars, 3-Day Conference & Masters' Conference	Hotel, Conference Center	$90-$300, $250, n/a	15-90, 300-350 (1000 through free Vendor Show), 30-40	Must be ITX member to join. Masters' Conference is a closed meeting for the "movers and shakers" of the HP user and vendor community; attendance by invitation only. Winner of 1989 Golden Scroll Award for newsletter. Has office with paid staff person and HP3000/37. Local group meetings held throughout year.
SERUG				2	1-Day Conference			60-100	
SIERRA-RUG	Individual	Free	50	4	Evening	Restaurant, Member's Site	Free	15-20	Also part of the 2-day NEVCAL Conference.
SLRUG				6	½-Day	HP Office	Free		
SMUG	Individual	Free	180	6	½-Day	HP Office	Free	20-60	Golf outing with Fred White summer, 1991.
STRUG	Individual	Free	15	4	½-Day	HP Office	Free	15-20	Also part of the 2-day All-Texas Conference.
SPORUG	Individual	$35	80	10	2-4 Hours	Restaurant	$7	25	
SVRUG		Free						25	

Appendix: North American Hewlett-Packard Users Groups

(Group Name, Area/Equipment Served, Newsletter, & Scholarship Information)

Short Group Name	Full Group Name	Geographic Area Served	Equipment Served	Affiliated with INTEREX (ITX) (1)	Year Formed	Form of Government	Newsletter (Issues per Year)	Pages per Newsletter Issue	Scholarship (Amount Awarded) (2)
TENN-RUG	Tennessee Regional Users Group	Tennessee	3000	Yes					
Toronto TUG	Toronto Technical Users Group	Toronto, Ontario, Canada		No					
TUG	HP3000 Toronto Users Group	Southwestern Ontario, Canada	3000	Yes	1979	Officers	None	None	None
Upper Midwest TUG	Upper Midwest Technical Users Group	Iowa, Minnesota, North Dakota, South Dakota, Wisconsin	1000, 9000	No					
USNY-RUG	Upstate New York Regional Users Group	Western New York (excluding Buffalo)	3000	Yes	Late 1970s	Board	None	None	None
WBTUG	Washington-Baltimore Technical Users Group	District of Columbia, Maryland, Northern Virginia, West Virginia	1000, 9000	Yes	1980	President	6	3-5	None
WNYRUG	Western New York Regional Users Group	Western New York	3000	Yes	1977	Board	None	None	None
WORUG	Western Oklahoma Regional Users Group	Western Oklahoma	3000	Yes	1987	Steering Committee	None	None	
WTUG	Wisconsin Technical Users Group	Wisconsin	1000	No					
WRUG	Wisconsin Regional Users Group	Wisconsin	3000, PC	Yes	1982	Officers	None	None	None

Notes:

(1) ITX is used as an abbreviation for INTEREX throughout the table.
(2) All dollar amounts are U.S. Dollars, except as noted.
(3) All conferences have a vendor show as part of the conference.

Appendix: North American Hewlett-Packard Users Groups
(Membership, Meeting & Other Information)

Short Group Name	Types of Member-ship	Annual Cost of Member-ship (2)	Number of Members	Meetings per Year	Type of Meeting (3)	Location of Meeting	Cost to Attend Meeting (2)	Typical Attend-ance	Other Comments
TENN-RUG									
Toronto TUG									
TUG	Individual	Free	200	6	5 ½-Day & Full-Day	HP Office, Hotel	Up to 3 ITX members for $20, Non-member: $25 each ($CD)	50-100	
Upper Midwest TUG				6					
USNY-RUG	Individual	Free	130	4	Full-Day	Hotel	ITX member: $15, Non-member: $25	30	To get into lunch at a meeting, you must give two ideas for future meeting topics. Has obtained Not-for-Profit status with the IRS.
WBTUG	Individual	Free	150	6	1000 & 9000 each ½-Day	HP Office	Free	1000: 7-10, 9000: 10-15	Each subgroup holds separate meeting in morning or afternoon, with joint meeting at lunch. Distributes three versions of newsletter: HP1000 only, HP9000 only, and combined. Winner of 1989 & 1990 Golden Scroll Award for newsletter. Has an HP1000 CSL.
WNYRUG	Corporate	$25	60	6	4 Evening, 2 picnic/ party	Restaurant	$17-$22	35-40	
WORUG				6	Evening			25	
WTUG									
WRUG	Individual	Free	180-200	2	Full-Day	Hotel	Free	75	

Bibliography

Amos, Diane — "Assessment of Value of RUGS to Users, Vendors and Hewlett-Packard", *INTEREXPress*, January, 1991, page 14.

Hart, Christopher — "Commentary: What is the purpose of a Regional Users Group?", *IMRUG Newsletter*, March, 1991, page 4.

Indiana Regional Users Group, Inc. (INRUG) and INTEREX — INRUG/INTEREX promotional brochure, April, 1991.

INTEREX — *Starting a Regional Business/Technical Users Group Guide*, May, 1991.

TITLE: HP 3000 Backup Strategies

AUTHOR: David Merit

ORBIT Software

319 Diablo Road
Suite 218

Danville, CA 94526

(415) 837-4143

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 6103

TITLE: 'L' Plates for IMAGE

AUTHOR: Shelby Robert

Proactive Systems

Four Main Street
Suite 101

Los Altos, CA 94022

(415) 949-9100

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 6104

Paper #6105

Bloodless Software Prototypes and Purchases

by
James Alexander

Dynamic Information Systems Corporation
652 Bair Island Road, Suite 101
Redwood City, CA 94063
(415) 367-9696

During the Vietnam War, the life expectancy for a Med-Evac helicopter door gunner was less than 30 seconds when performing duties in a battle. Stressful? We can assume it was! The most dangerous occupation in America today is a firefighter. Stressful? Let's just say, it would not be a good occupation for someone with high blood pressure! And then we have the combination of the two positions: the role of the MIS manager!

The MIS manager today is constantly fighting fires all while maintaining a position that has the exposure for anyone to take potshots at them. MIS Managers suffer from some circumstances common to all professional management positions: an overburdening of high priorities, a lack of understanding from upper management, confusion of tools available, and a lack of time to deal effectively with the first three items just mentioned.

Keeping this in mind it is easy to see why staying current in the industry is nearly impossible. Those successful at it have found some simple procedures that allow them to remain sane. I will offer guidelines that allow for management of one critical area, the successful evaluation, prototype, and purchase of major computer software acquisitions and I provide some important recommendations for dealing with software vendors.

The first area ignored by most is Market Awareness. It happens on two levels: In the office and outside the office contacts. While most MIS managers get the current crop of periodicals, there are some who don't see any benefit. In the Hewlett-Packard marketplace we have some excellent options such as the HP Chronicle, Interact magazine from Interex, HP Professional, SuperGroup magazine, plus several publications from HP, other vendors and SIGs (Special Interest Groups). The second problem, which happens

more easily than not receiving the information is passing it on without properly reviewing the contents. When reviewed, the information will force one to take the next step to staying informed, requesting information on a new product or service. Without further examining a possibly useful new tool, we restrict ourselves to present methods of operation that we know are not working. If time does not allow for complete review, pass an article to a subordinate with instructions to summarize the information. This will allow review even if at arms length.

Create a file of information about products and services that may enhance your management possibilities. Add to and review this file on a regular basis. Request additional information for this file to be updated.

Another area of missed opportunity to keep informed of current events in the marketplace is through good communication with the HP Software Sales Representative or Software Engineer. This can be an effective method of finding out about new developments within HP and by third party vendors. HP support personnel work with many different sites which exposes them to a great variety of new and useful tools. It can be useful to employ their experience and knowledge.

Outside the office is a world of information and ideas waiting. These opportunities will not come to you but can be had with a small amount of effort.

First, become involved with local user groups. This is usually a low cost or no cost method of networking with other people in the HP market place. It allows for sharing knowledge and experiences on an informal level without large time commitments.

Second, get involved with peer group organizations. For example, if you are a MIS manager in a school district environment, become involved in the local, state, and national organizations for Educational Data Processing. There are organizations for many different industries. This will give you exposure outside your own world (Hewlett-Packard).

Third, become active in Special Interest Groups that can fulfill an interest you have. Some examples, SIGIMAGE or SIGSYSMAN, both offer excellent opportunities to get important information from other users experience and tips on good management of systems and resources.

Fourth, take part in an on-going educational process. This may be in specialized seminars offered by private organizations, courses at a local community college or university, or vendor education classes or seminars. This will expose you to new opportunities and individuals who can widen your knowledge base and enhance your solutions toolkit.

Lastly, attend regional (BARUG, NOWRUG, SCRUG) and national (Interex) user group meetings including vendor trade shows. The technical and managerial sessions at these events can supply complete solutions to your immediate problems. You can speak directly with experts in all areas of Hewlett-Packard systems and applications. You can present your solutions to the other members by developing papers to be presented and gain recognition as an expert in your field.

By attending the vendor trade shows you have some major advantages but may be held back by certain fears and trepidations. Those fears may include talking to vendors you have said no to in the past, assuming hard sell tactics will be used, or expecting that spending time in vendor display area is a waste of time. These fears are not justified when examined: Talking to a vendor you said no to before might be uncomfortable at first but you will quickly realize it is not a personal issue and you probably have the respect of the vendor for taking action rather than just ignoring the issue. Assuming hard sell tactics will be used at the show, this is handled more easily at the vendor show because you can simply walk away. Lastly, seeing the vendor show as a waste of time is easily removed when the benefits are examined.

Advantages of attending a vendor trade show are numerous. First, you learn about products. You can find out about new products on the marketplace or updates to products you have looked at before. Perhaps that product you had to turn down now has the major feature that was your necessity before considering a purchase. You can find out about enhancements to products you already own. Many people have new releases of products but have not implemented them because they weren't sure of the benefits they would receive, yet they find out about those important items at the Vendor booth. You have the opportunity of seeing a live demo of the product when you stop at the booth. Vendor availability allows you to do a product to product comparison and get immediate answers to questions that come up during the comparison.

Collect literature and product information even for products that may only be a future need. This means later you have product information you can request updates of when the need arises.

You can identify which sales representative for which vendors you prefer working with. While sales representatives normally have specific territories, a smart vendor will make an exception if you specify you prefer working with another sales representative.

Arrange for onsite visits for products that seem to present immediate benefits and take more time in off show hours to discuss benefits & feature with vendor representative. Some vendors will have trial tapes you can take with you or you can sometimes request a trial at no cost at a vendor show when normally there is a trial fee.

You may have opportunity to participate in a special show pricing when you start the evaluation process of a particular software at the vendor show. This can amount to a significant dollars savings for your company.

Gather ballpark figures that you need for budgeting purposes. Knowing these amounts is essential because you will be requested to provide your estimates of budget needs and you cannot plan for expenditures without knowing how much you may be required to spend to purchase the solution.

Use the vendor show to meet the management personnel for the vendor. This could be helpful if problems or disagreements arise later. The quality and commitment of the overall company can be measured by the quality and commitment of the management personnel.

At the vendor booth, take the opportunity to meet and establish communication with technical support staff to judge quality of support and ability to understand your needs and problems. You will be relying on these people to help you in crisis and this could be critical to your purchasing decision.

You may be in a wait mode or be planning for development in the future. If so, meet the vendor early but make them aware it is early and also indicate when would be the appropriate time to discuss needs to be filled. This will establish with the vendor the need to keep you in mind as new products develop and keep you up to date with the current state of the software you are interested in. You will be informed when a move is necessary on your part.

All of the above will prepare you with the knowledge necessary to meet the challenges facing you in MIS management of your company. The next most important step you must do is stay on top of the corporate or company goals and objectives. You must form a requirements list and an action plan to support the company goals. You will have the proper information to offer possible solutions without the need to do any large amount of research because you have keep abreast of currently available solutions. You can immediately start prioritizing the solutions and seek additional information to fulfill those needs.

With this knowledge at hand, you can develop a Request for Proposal if you use that process and submit it to the appropriate vendors for responses. This will establish with the vendor you are serious and have given this thought.

This begins the true interaction with the vendor. First meetings with vendors will generally be one of meeting and establishing a working relationship with an exchange of necessary information. Give the vendor as much information as you are able. Explain the overall functions of your business operations and how MIS support is utilized. Share an organizational chart with the vendor representative so they may understand your relationship with those you support and who will be involved in your upcoming acquisition. Insist on the vendor representative meeting with the department heads affected by the software purchase.

Keep the vendor well informed and expect a good amount of communication between you and the vendor representative during the software trial period. Let the vendor know promptly both the good results as well as reporting the negative aspects of the trial. This will show mutual trust between you and the vendor. If you believe vendors are in the business only to sell you a product, you are mistaken. There is no denying vendors are in the business of staying viable and profitable, they are also in the business of gaining your respect and trust so you'll refer them to other accounts that could use what they have to offer. Treat a vendor they way you would like to be treated and you will establish the groundwork for a long term relationship. Learn as much about the vendor as possible. Find out about reference customers who can show you an ongoing relationship with the vendor. Expect reference accounts in the same geographic area and obtain some in the same or similar industry as your own.

Obtain a written estimate for the software quoted. Accept verbal price quotes with reluctance, understanding until it is committed to writing, price or what is included may change. Expect a detailed quote showing all that is included. Is first year maintenance included or must it be added later? Does this quote include training, consulting, or implementation assistance? Are there any discounts to be applied or extras to be added on? What time period is the quote good for? These may seem to be trivial but they are details easy to miss.

Be willing to agree to a non-disclosure statement to allow access to more detail information or future direction of the company. You may find the vendor plans for enhancements that meet your exact needs or is planning a course of development that will save you large amounts of effort in the future.

You should now be in a position to establish objectives for a trial of the software that meets your needs. Utilize the vendor representative heavily as they have had experience in the past that has lead to successful trials of their software. You might argue this leans the trial toward purchase, but you must understand the vendor is there to help you do the trial not to blindly lead you to a purchase and that would require eliminating the trial if at all possible which is not what the responsible vendor will do. If you are buying shoes, you accept the foot measurement when done by the salesman, the vendor representative is doing nothing more than showing you how the software will fit your size.

Make sure to budget enough time for the trial period. Most vendors have estimates about how long the trial should take. Use this as a starting point at least. Add time where you see fit. Always assign personnel up front and accept the impact on the present workload. After all, we are working at having a successful trial and that requires consuming some resources. If you find a short fall of resources or an emergency arises during the trial process, apprise the vendor of the situation and determine an estimated date the trial can resume. As early as possible during the trial phase, determine if you require a prototype to be done as purchase justification for upper management. Let the vendor know as they will need to allocate some resources for your prototype and this can be planned.

Examine and exercise all features indicated for the product. Not all features will be important to you but you should at least understand why they are there. For those features that you find critical to your purchase decision, spend in-depth time and concentrate on those product qualities. Don't get hung up on features that have no meaning to you. Some people will not understand or need certain features but will waste time testing and questioning these items.

When it is appropriate, get end users involved early in the trial process. They see needs from a different perspective and may bring up important factors you did not consider. They may also pose objections you did not identify.

Ask the vendor about user group meetings in your area. One of the best assets is other users of the product. If you can get together and discuss your situation with like minded people, you can make a much more informed decision.

Ask the vendor to organize an end user visit with another company using the product. Select the appropriate personnel from your end user community and schedule meetings with people of the same job function in the other company. Make sure you are knowledgeable of the product before you visit. Ask for a firm of similar size and computer configuration if possible. Check the accuracy of vendor statement, bug fixes, and technical support as this company has probably had experience in each of these areas. Find out what competing products were investigated and ask why this product was chosen over the competition. Establish contacts for later questions and develop friendliness with like end users. Ask about the quality of training.

This will probably have concluded your investigation and trial stages and you will either be at a purchase decision or will have decided to develop a complete prototype for purchase justification. If you will be purchasing at this point and many people do, you will want to ignore the next few paragraphs as they will deal with the prototyping process.

When starting a prototype, as when starting a trial, you must make sure you develop time lines for the prototype. You need this for the vendor to provide the necessary technical assistance and for you to allocate the in house personnel.

Establish the decision criteria for the purchase only after adequate initial analysis during the prototype. This allows establishing measurable objectives during the prototype and will help in documenting the purchase justification to upper management. Give the vendor your purchase decision criteria as early as possible.

Expect the vendor to provide as much technical assistance as you require during the prototype. This is an excellent time to observe the technical competence and support you can expect in the future from the vendor. You can assume you will receive the highest level of technical service during your prototype phase. While most vendors technical support will continue at this level, if there is any doubt during the initial prototype phase about the technical assistance you receive, you must know this is an indication of unacceptable ongoing support. Utilize the vendor assistance to develop code and

structure that relieves your staff from ramp up time on new software but does give them examples of how to approach future projects and development.

With technical help from the vendor, find the best fit areas for the proposed software. Accept the judgement and caution of the technical support when indicating areas where the software is not needed and will not help. Understand the reason for this and accept the limitation indicated for the product. Good technical support will be up front about the best places to use the software, will guided you toward those, and will identify pitfalls for you to avoid in the future.

Focus prototype activities in areas where the software could be used in production immediately when the decision to purchase is made. This will expose the software to end user requirements as soon as possible and may uncover unforseen problems before the warranty period expires.

After completing the prototype, work with the vendor to develop the cost / benefit analysis for presentation to upper management if necessary. The vendor may have samples of other cost / benefit justifications from other customers that will reduce your required effort and speed the process along.

Assuming the purchase decision is made you now can start negotiating with the vendor representative. Some companies allow more bargaining room than others during this period. Don't be shy in asking for extras and amenities. Ask about options available. It may behove you to acquire an option now because of a significantly reduced price when bundled in the original purchase. Don't forget about adding training into the purchase price, it is easier and sometimes cheaper to buy training when the software is purchased than to go back to the money table for additional training dollars. If you see a machine upgrade in the near future, ask about the savings for purchasing for the new machine in the original order agreement. Establish a firm delivery date for the new software. Identify when renewal dates for maintenance occurs, you may want to adjust these dates to occur in conjunction with you company's fiscal calendar. Finally, make the purchase decision, get the agreement back to the vendor and you will be finished with the process and satisfied with your efforts.

After the purchase, utilize the vendor technical help whenever necessary. Do not hesitate to call for help when stuck. But don't cry wolf, make the call for technical support a valid call. Identify to the support staff when the call is critical and when it is not. Expect the critical calls to receive the proper attention. Let the vendor know when technical support

is good or bad. They are there for your support but cannot improve unless you make the proper people aware of your difficulties with support calls. Also, when you can compliment the support staff you become known and will receive extra attention when calling.

Get involved in the products users group meeting. Also, if a SIG exists, be involved and active. Sharing your knowledge and experience in these groups will return to you many benefits and will help you gain stature in your peer groups.

Offer to become a reference customer for the product. Offer to do a beta test of new products and new releases of software. Refer customers to the vendor representative when you know another customer who could benefit from the products. Look at the vendor as an extension of your company, treat them like you would like to be treated. Offer suggestions for new features or products. These customers receive special attention and can expect additional attention from the vendor.

The end result of implementing these techniques is felt by three entities.

First, the vendor will have a successful placement of the software product. The honesty and trust built during the acquisition process with continue to strengthen. New products from this vendor may assist in easing the burden of management further and the vendor will keep you informed of new developments and enhancements.

Second, your company will benefit from better utilization of your computer resources. The end user community will be more productive and can concentrate more on the job at hand than be slaves to the old way of doing things. The company will see payback happen and eventually save money as a result.

Last but most important, your personal workload will be lessened. You will be recognized as an efficient and enlightened MIS manager. The end users will be happy with your efforts, and you will be satisfied with a job well done.

Old MacDonald had a network

Anthony C. Furnivall
SDL/Software, Inc
P.O. Box 601
BUFFALO NY 14222

When first confronting the world of networks, the stoutest heart trembles at the jargon that is used so freely. This presentation will demonstrate how all the talk about protocol stacks, layered architecture and so on really does have a useful part to play in getting the job done. It involves members of the audience and demonstrates through such participation just how to make sense of the barnyard babble!

INTRODUCTION

I can imagine few things more daunting than walking into a networking show, with little or no idea of what the vendors and attendees are talking about. Where else can you hear such a density of meaningless TLAs (three letter acronyms) flying around with such abandon? Trying to make sense of the jargon is a tough prospect. My goal in this paper is to provide something that you can read and use as a reference either before or after (or even in stead of!) attending the actual presentation.

It has been said that the mind retains 10% of what it hears, 20% of what it sees and 50% of what it does, and for this reason the main thrust of the paper is a participatory experience. However, not every one can attend, and even for those who do, some follow-up might be useful.

What follows, then, is a plain and easy introduction to the idea of network layers, and network protocols. It is not intended to be a reference on the subject - indeed, there are several places where accuracy has been sacrificed for the purpose of clarity. However, it should stimulate you into going on and doing more reading.

NETWORK LAYERS

It is almost inconceivable that at some point during the past 12 months, you have not seen a diagram like Fig. 1. This diagram shows what is known as the ISO OSI model. I prefer to think of it as 7-layer network cake.

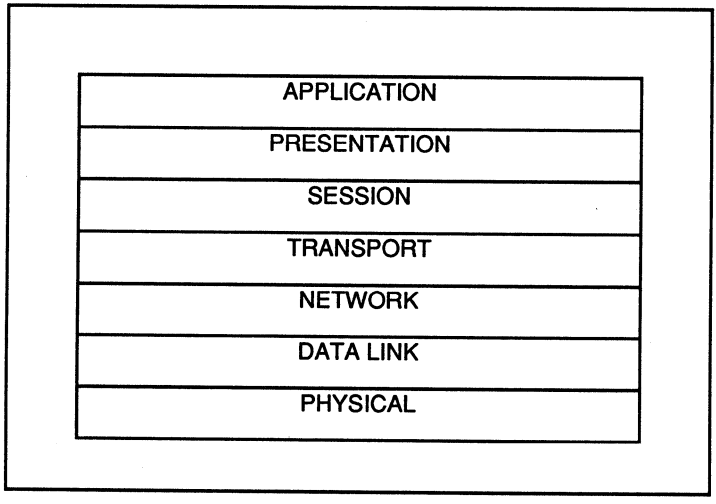


Fig. 1 - 7-layer network cake

This innocent little picture conceals a wealth of information, and a rich potential for confusion. The next few paragraphs will attempt to give a simple idea of what the various layers do.

Why do we need layers?

The first, and most important question to be answered is, Why do we need layers? What is it that is special about the layers in a picture like *Fig. 1* ?

The answer lies in the fact that it is easier to divide a complex task up into small pieces, and then to manage the small pieces individually. This is a common feature of software engineering, people management or almost any other field of human endeavour. No surprise, then, that the same thing holds true with computer networks, or data communications.

To define the entire set of tasks in one complex document would make it almost impossible to connect any computers together. What has happened, therefore, is that the task has been broken down, into reasonably logical pieces. These pieces represent the layers of the International Stack Of Protocols.

Physical

The physical layer is concerned with how the data is represented in terms of voltages and frequencies. It also deals with the size and shape of the connectors used to plug in to the network, and the types of cable used to carry the signals. Pretty boring stuff, for the most part, but very important. If you have ever tried to plug a three-pin plug into a 2-pin socket, you can imagine how frustrating it is not to have the right connections for a network system. Especially since the little adapter that you get for 79¢ at K-Mart costs several thousand dollars in the network store!

What we have, then, at this layer is the ability to create a 0 or a 1, several times a second (actually, several million times a second, but what's a few orders of magnitude between friends).

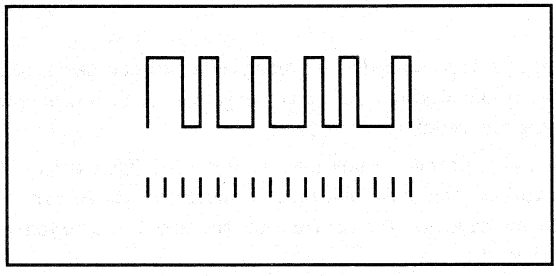


Fig. 2 - Signals varying over time

Data link

The data link layer is concerned with how the signals of the physical layer (the '0's and '1's) change over time. A voltage change is used to represent a data value. You may like to think of the options available at this level rather like the difference between AM and FM radio.Both types of radio are useful for bringing information into the home, but they are not directly compatible.

Some of the buzz words that you will hear in connection with this layer are terms like Ethernet, Token-Ring and MAC layer. We will explore Ethernet and Token-Ring a little later. MAC stands for Media Access Control, and refers to a common standard for moving data out onto the cables of the physical layer.

A sequence of bits, at this layer is called a frame. The bits have specific meanings, and these meanings are described by the various protocols which operate at this layer (such as Ethernet and Token-Ring). In addition, the bits are grouped together, generally in groups of 8. However, reflecting the truly catholic background of networking, these groups of 8 bits are called not a byte, but an octet. There is a good reason for this, but it can be confusing for beginners, especially if you have an interest in music, and regard an octet as a piece of music for 8 players!

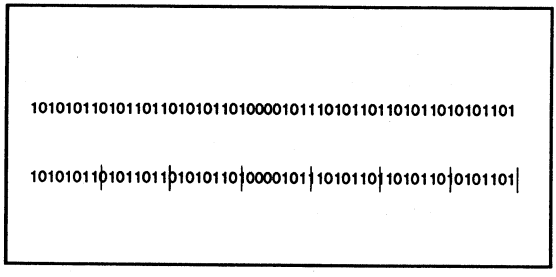


Fig. 3 - A septet of octets (56 Kinda Binary Signals)

Network

The network layer is concerned with making certain that the data is delivered to the right place. This is rather like making certain that you get your mail, and no-one else's. There are several interesting features of this analogy.

Just as much of your mail is junk-mail, addressed to 'OCCUPANT', so much of the data on the network is potentially junk-mail. It actually has rather more value to someone, but a lot of the messages on the network are addressed to everyone, in the hope that someone will respond.

The second interesting feature concerns the way in which the mail (read network messages) gets delivered. Rather like the Post Office, there is no guarantee that the mail

will ever get through, or if it does, that it will get through in the right sequence, or if it does that it will all get through. This is rather like ordering a new set of tools from Sears, and receiving the bill before the tools!

Thirdly, just as in the mail system, the data is now called a packet. (What started off as a bit, in the physical layer, and became a frame in the data link layer has now changed its name yet again. However, we are still talking about bytes (or octets).). Like most pieces of mail it has an address to which it is (hopefully) delivered, and a return address from which it came.

Finally, just as much mail is local in nature, a large amount of it travels to different ZIP codes, or different states, or even different countries. The address on the packet indicates exactly where it has to go, but different parts of the mail system read and use different portions of the address information.

The network layer is like the infra-structure of the postal system. It takes care of finding the destination for the packet in question, and finding the best way to deliver it. However, it does *not* represent or include the additional intelligence that users of the Postal System need. This intelligence is contained in the next layer.

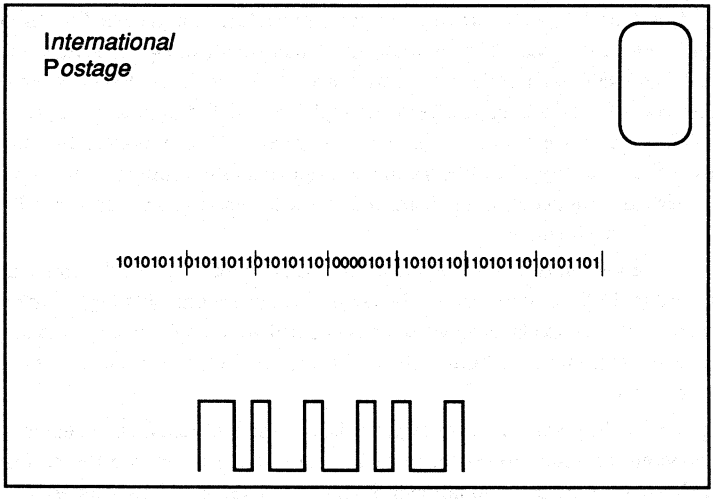


Fig. 4 - A simple network packet – *Truly Child's Play*

Transport

At this point, I have to make a confession.

On various occasions in the past, I have been late making a payment on a credit-card. What happened was that the credit-card company sent me a bill, and I guess it got mislaid, or otherwise ignored. In their kindness they sent me another, and another. Pretty soon,

though, they resorted to a different kind of network system, and started calling me at all sorts of hours of the day and night!

What they were doing, in effect, was adding just that intelligence that is missing at the network layer. They had a reasonable expectation that the message they sent me (the bill), would result in a response from me (also via the mail). When this did not happen, they started off by sending out a copy of the bill. Eventually they realized that nothing was going to happen, and changed their tactics.

This is the function of the transport layer. It exists to guarantee a reliable, correctly sequenced stream of data between one node on the network, and another. It includes provisions to keep track of the number of packets in the stream at any time, and to let the sender know if any of them were received out of sequence. (Remember those tools from Sears).

Session

:HELLO!

What happened to our easy introduction to networking protocols. If you glance back again at Fig. 4, it doesn't seem a particularly simple procedure to understand so far. The fact is, that in terms of the plain old terminal systems that we all know and love, we have only created the equivalent of a serial cable between an ATP port, and a VDT. (Watch out for those acronyms - I should have said between an Advanced Terminal Controller, and a Video Display Terminal). The reason for this extra complexity is because rather than having a fixed, permanent, dedicated circuit between the terminal and the computer, we have an open, shared, virtual circuit connecting them, and all the lowest layers are concerned with is managing that virtual circuit.

The :JOB of the session layer is to begin the process of interacting with the computer at the end of the cable. MPE provides session layers services every time you log on, or every time you stream a job. Session layer services are concerned with providing access to data in a controlled way, with accountability. These services include file security, resource accounting and so on.

This is generally the point where you begin to heave a sigh of relief, and begin to have something concrete to which to relate. Indeed, the top three layers are the easiest to understand. However, it helps to go through them to get a sense of how networking can complicate things.

On a VAX system, for example, you don't say :HELLO, you say $LOGON. I have sat in front of a VAX system and tried to persuade it to respond to MPE commands, with very limited success. Fortunately HELP was one of my few successes! When we are tying together different computers with a network, some provision has to be made for a common sequence of commands to gain access to these session services.

Finding out the names of all the files available to you, for example, might involve using any or all of the following commands:

- LISTF
- FSTAT
- DIR

The session layer is responsible for making access to the computer as easy as possible.

Presentation

If we type LISTF on an HP system, we expect to get a list of information about a set of files. However, this information is not stored directly in the same way that it is presented to us. What we have is a software routine (think 'layer') which takes data in one format, and transforms it into another, for presentation to the user. (Incidentally, if you think that a user is soon going to be called an application, you're right!).

Other presentation services include such arcane things as making certain that the bytes which make up a 2 or 4 byte structure are presented in the right sequence, and making certain that the data structures which are expected at a certain point are indeed that data structures that are provided. The problem of byte sequencing is also known as byte-gender, which just goes to show that sex rears its ugly head in all sorts of strange places!

This is very important, however, because if I have a value such as 123456 on an HP machine, it could become any of the following, depending on the byte gender of the machines involved:

-499122175

1088553216

16793826

As you can imagine, transferring funds from a bank which runs HP to a bank which runs, say, Tandem equipment could result in some very strange accounting unless the presentation layer is firmly in place. The same is true, incidentally, for other very slow networks, such as magnetic tape.

I recently completed some data conversion for a client where they were moving data between two systems, and one of the was a VAX. I was doing the conversion on an HP, and sent of a tape with all ASCII data (I was smart enough to avoid problems with binary formats, because of the byte gender differences), but we discovered that the HP and VAX systems represented signed ASCII digits differently. This was my first (and hopefully last) introduction to the Signed Separate clause in COBOL!

Application

The application layer is the layer which provides real services and results to real users. Sometimes this can be an interactive user, and sometimes it can be just a program somewhere which wants to reach out and touch someone one the network. Among the sort of things which you can find happening at this layer are file transfer, terminal connection and network management.

Once we reach the Session layer, the services that are provided are very tightly interlinked. This is in contrast to the lower four layers, where it is quite possible to mix and match. The freedom which this leads to can be confusing because it is important to make certain that both ends of a conversation are speaking the same sort of language.

PROTOCOLS

Our brief and somewhat unguided tour through the 7-layer network cake has touched only briefly on what the various layers accomplish. The purpose of protocols is to determine *how* they accomplish it.

A protocol is defined as:

- The customs and regulations dealing with diplomatic formality, precedence and etiquette.

What we're going to do next is to take a quick look at some of the protocols which are popular at each of the three lowest levels.

Physical layer protocols

At the lowest layer, the physical layer, there is not really much that can be called a protocol. The way in which most networks get data on and off a piece of wire is a rather strange technique called Differential Manchester Encoding. All you really need to know about this is that it works, and provides several additional benefits for the guys next up the line, the data link layer.

Data link layer protocols

The big players in the data-link league are Ethernet, along with its cousin IEE802.3, and the blue-flavoured Token Ring. This is where the first of the crusades is waged, with people arguing in favour of one approach, and other people taking a contrary point of view.

Ethernet and IEEE802.3 are virtually identical. Unfortunately, from a genetic point of view, so are a man and a dog. In both cases, the species can co-exist but communication between them is of limited functionality. Ethernet also has a poor foster sister called Localtalk, which is used to tie together Apple Macintosh computers, and is based on the same concepts as Ethernet. These concepts are known as Carrier Sensing with Multiple

Access and Collision Detection. This mouthful is more frequently known as CSMA/CD, and is a notable departure from the rule that complicated networking terms have three-letter names.

What happens here is that anyone can talk on the network at anytime (Multiple Access), just so long as no-one else is (Carrier Sensing). When they are talking on the network, they keep listening to see if anyone else started around about the same time (Collision Detection). If so, both parties stop, and wait a bit before trying again.

This is much easier to demonstrate than to describe, and is the techniques used by most people in most walks of life, most of the time. (Yes, there are exceptions, in the computer world they are referred to as jabbering nodes, in the real world they are called boors).

In contrast to CSMA/CD, the other common way of controlling access to the network , is to pass the pipe of peace. This is like the way in which speakers in Congress are scheduled (I never promised you a reality based example!), and guarantees that there is an orderly progression of opportunities to speak. Each node on the network gets a special token, and can only access the network when it has the token.

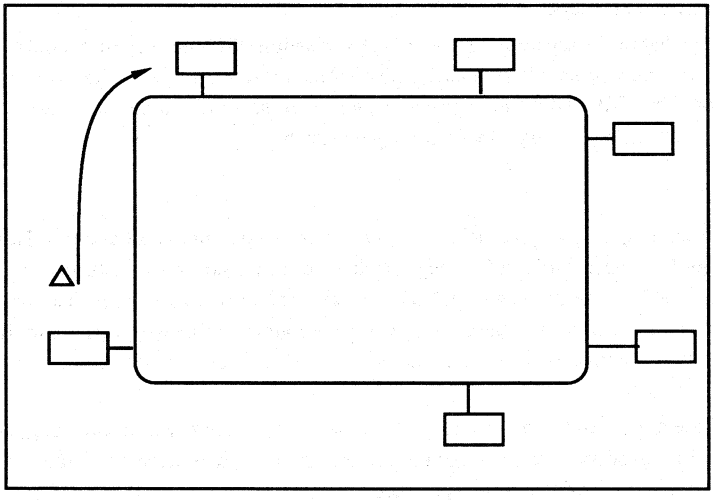


Fig. 5 - One-way token rotation

From a technical perspective, Ethernet is a non-deterministic protocol, and Token Ring is a deterministic protocol. This means that we can not predict who will talk next with Ethernet, but that no-one is unduly delayed, while we always know who is going to have the token next, but we may have to wait for it to go all the way around the ring, if we are busy.

Probably only the Super Bowl or the World Series generate more heated discussions and arguments about the merits of opposing teams than these protocols. The truth is that both work, and work well. Either network protocol does the job, but anyone who wants to, can ring the changes on the old arguments. Most people just get on with the job of implementing one or the other.

Network layer protocols

Network layer protocols, you remember, are concerned with making certain that the correct recipient of a packet can be identified. This involves not only tracking them down, but making certain that you can remember where to find them later on!

The most common protocol in use at this layer, is a protocol called the Internet Protocol, which is almost always seen with its buddy, the Transmission Control Protocol. These protocols were developed for use in the Department of Defense Advanced Research Projects Agency (ARPA) network, and have become a de facto standard. Everyone claims to be able to deliver a TCP/IP implementation, and this is one of the easiest ways to hook up different computers.

Now that the International Standards Organisation (ISO) has gotten in on the act, there are several new protocols available which match more closely the other standards being promoted by ISO. All of these protocols, however, serve the same basic purpose, that of making certain that the mail does indeed get through.

SUMMARY

Networking is a very complex subject - there is no doubt at all about it. However, it need not be unnecessarily off-putting. While some of the concepts are strange at first, they are also self-evidently useful, and serve clearly identifiable purposes. The best way to become more familiar with the language and environment of networking, is simply to do it. Attend training classes, read as much as you can, and put your learning into practice as much as possible.

There are several opportunities to become more familiar with networking, and while some of them seem to exclude beginners, almost all of them have something of value to people with varying degrees of experience.

QUIZ For Beginners (And Not So Beginners)

Robyn Kerekes
Boat America Corporation
880 S. Pickett Street
Alexandria, VA 22304
(703) 823-9550

As anyone who has ever written a QUIZ program knows, it is extremely easy to learn the rudiments of this report generator. A little thought, a few commands (i.e., ACCESS, REPORT, and GO), and you have your report. The syntax is simple to remember, the programming concepts are straight forward and, all matters being equal, it is easier to produce a report using QUIZ than it is using a third generation language such as COBOL or RPG. Unfortunately, all matters are not equal. Even for simple reports, QUIZ can consume an enormous amount of time and system resources. This can be more than a mere nuisance to the person running and waiting for his QUIZ report to appear. Other users on the system may find the performance of their applications slowed significantly as QUIZ grinds its way towards completion. In addition, there are those times when it seems that you just "can't get there from here," that is, it doesn't seem possible to produce the report that you want using QUIZ. This paper is written for the QUIZ programmer who learned to program by reading, or leafing through, the manual. It addresses QUIZ program performance and provides some application tips to help you "get there from here."

PERFORMANCE

The easiest way to think about improving performance is to think *reduce*: reduce the number of records read, reduce the number of records selected, reduce the number of decisions QUIZ has to make. Simply put, the less data QUIZ has to process, and the less processing QUIZ has to do of this data, the less time (and CPU) QUIZ will need to generate the report. Two of these areas, retrieval and selection, are the responsibility of the ACCESS, CHOOSE and SELECT statements. They act in conjunction to build the record complexes needed to produce the report. A clear understanding of how QUIZ uses these two commands to generate record complexes is essential when undertaking to improve program efficiency. Below is a simplified outline of how QUIZ retrieves and selects data when building record complexes:

- a record from the primary file is read (if CHOOSE is present, these records are read by key value, otherwise they are read sequentially);

- DEFINE statements based upon the primary file are evaluated;

- SELECT conditions for items found in the primary file are evaluated (first, those conditions found in the SELECT file IF command, then those found in the SELECT IF command);

- QUIZ LINKs the primary file with the subordinate file via a key value in the subordinate file if the selection criteria was met;

- DEFINE statements based upon the subordinate file are evaluated; and

- SELECT conditions for items found in the subordinate file are evaluated (first, those conditions found in the SELECT file IF command, then those found in the SELECT IF command).

This outline brings several critical points to light. The first of these is the importance of the primary file in the ACCESS statement. Try to reduce the number of records read by making the file which is most likely to exclude data the primary file. The most obvious way to do this is to designate the smallest acceptable file to be the primary file.

EXAMPLE: Assume two files, cust-mast, with 100,000 records, and pay-dtl, with 10,000 records. If you want a report of only those customers with information in pay-dtl:

```
>ACCESS pay-dtl LINK TO cust-mast
>REPORT ...
>...
```

QUIZ reads only those records in cust-mast which have a corresponding record in pay-dtl, at most, 10,000 records. If the linkage was reversed, i.e. ACCESS cust-mast LINK TO pay-dtl, QUIZ would read all 100,000 records in cust-mast in its effort to determine which customers have a record in pay-dtl.

Another way to exclude data and reduce the number of records read is to make the file which is most likely to fail your selection criteria the primary file. If the record from the primary file fails the selection process, QUIZ will discard this record and, subsequently, not attempt to link to and retrieve any records from the subordinate file.

EXAMPLE: Assume, in addition to the files in the previous example, a third file, cust-dtl, containing 100,000 records.

```
>;Inefficient Code
>ACCESS cust-mast LINK TO cust-dtl
>SELECT IF last-trans-date of cust-dtl > 910101
>...
```

```
>;Efficient Code
>ACCESS cust-dtl LINK TO cust-mast
>SELECT IF last-trans-date of cust-dtl > 910101
>...
```

In the inefficient code, QUIZ must read through each record in cust-mast, link it to and read a record in cust-dtl, and then evaluate the SELECT statement. In the efficient example, QUIZ will read a cust-dtl record and then evaluate the SELECT statement. Only if the cust-dtl record passes the selection criteria will QUIZ attempt to link to and read a record from cust-mast.

This logic should be applied to subordinate files also. Reduce the number of records read by placing those files which are most likely to exclude records early in the file list.

EXAMPLE:

```
>ACCESS cust-mast LINK TO cust-dtl LINK TO pay-dtl
>SELECT IF pay-type of pay-dtl = "C"
>...

>;If pay-type is seldom equal to "C", a more efficient
>;solution is:
>ACCESS cust-mast LINK TO pay-dtl LINK TO cust-dtl
>SELECT IF pay-type of pay-dtl = "C"
>...
```

Another critical point in building record complexes which has a tremendous impact on performance is the manner in which QUIZ processes the CHOOSE, SELECT IF, and SELECT file IF statements. Of the three, the CHOOSE statement is the most efficient, allowing you direct retrieval of specific records via a key value in the primary file. Unless CHOOSE is present, QUIZ will read each record of the primary file sequentially and processing time will be increased, often substantially. Because CHOOSE dramatically reduces the number of records read, it should be used whenever possible.

EXAMPLE: Assume cust-nbr is a key-item in cust-mast.

```
>ACCESS cust-mast
>SELECT IF cust-nbr = 123
>REPORT ...

>;QUIZ will read every record in cust-mast, but report
>;only those records where cust-nbr = 123. This is in
>;contrast to:

>ACCESS cust-mast
>CHOOSE cust-nbr 123
>REPORT ...

>;QUIZ will read only those records where cust-nbr = 123.
```

If your data structures and/or report specifications do not allow for CHOOSE, you are left with a choice of SELECT file IF and SELECT IF. From the standpoint of efficiency, these two statements are virtually equivalent. I quote my QUIZ manual:

> The SELECT IF statement defines the selection condition that is applied against the files in the ACCESS list. If the record complex fails to satisfy the condition, it is bypassed and not processed further. The condition is applied, to the extent possible, as the record complex is built. As soon as the condition fails, building of the record complex stops and the next record complex begins. (Cognos, 1985, p.R2.29)

Additionally, the SELECT file IF statement

> applies the condition to the records of the named file as it is read by the ACCESS statement. If the condition is not satisfied, the record, not the record complex, is bypassed and the next record is read.

Simply put, SELECT IF requires no more processing than SELECT file IF. The two are not, however, equivalent in all aspects. They will produce vastly different reports when your file linkage is parallel. Use the statement you are most comfortable with that will correctly select the data needed for your report.

A third area which has an impact on performance is defined items. The process of defining items is performed before records are selected. If you have numerous defined items and a large number of records to read, it may be more efficient to use two passes to create your report. The first pass selects records and reports them to a subfile. The second pass incorporates the defined items and actually produces the report. Notice I said "may be more efficient." This method should not be used if most of the records read will meet the selection criteria.

EXAMPLE:

```
>ACCESS cust-mast
>SELECT IF cust-type = "ACTIVE"
>DEFINE item1 = ...
>DEFINE item2 = ...
>; ... for a total of 25 defines.

>;The 25 DEFINEs are performed 100,000 times, once for
>;each record read in cust-mast.  A more resourceful
>;solution is:
>ACCESS cust-mast
>SELECT IF cust-type = "ACTIVE"
```

```
>REPORT SUMMARY ...
>SET SUBFILE NAME subfile
>GO

>ACCESS *subfile
>DEFINE item1 = ...
>DEFINE item2 = ...
>... for a total of 25 defines.
;The 25 DEFINEs will be performed once for each record in
;the subfile, which should contain substantially less
;records than cust-mast.
```

A final area to consider when programming for performance is the sequence of items used in AND/OR expressions, such as "SELECT IF item1 = value1 AND item2 = value2." When using AND, place the least likely condition first. In this way, QUIZ can reject the record complex without evaluating the remainder of the statement. Conversely, when using OR, place the most likely condition first. This enables QUIZ to accept the record complex without taking the time to evaluate the remaining conditions.

Remember, the goal is to reduce. Program so QUIZ will reject a record, or record complex, as quickly as possible, so it may move on to the next processing step.

There are numerous other ways to improve program performance which have not been addressed here. If you haven't looked through your QUIZ manual lately, do so. Pay particular attention to the section on performance as well as the discussions which follow the explanation of QUIZ statements. Before writing your programs, give the report specifications some thought, especially in relation to the data structures. Consider how the data "looks," what is likely to occur and what isn't.

APPLICATION TIPS

To aid the reader, I have included the following applications tips which make use of three extremely useful strategies. They are presented as cases with very little explanation.

CASE 1: Produce a report of all sales by sales person showing the sales person's name and sales dollars. If sales dollars are greater than zero, print sales dollars. If sales dollars are less than or equal to zero, print 'NO SALES'.

```
>ACCESS sales-mast
>DEFINE no-sales char*8 = "NO SALES"  &
>       if sales-dollars = 0 else ""
>REPORT &
>  tab 2 sales-name heading "NAME" &
>  tab 40 sales-dollars bwz heading "TOTAL SALES" &
>  tab 40 no-sales
>GO
```

CASE 2: Produce a report of all sales by sales person. Include the sales person's name and sales dollars, as well as the percentage of the individual's sales dollars to total sales dollars.

```
>ACCESS sales-mast
>DEFINE nbr = 1
>DEFINE total-sales = sales-dollars
>SORTED on nbr
>REPORT SUMMARY total-sales subtotal noreset
>SET SUBFILE AT nbr SIZE 1 NAME myfile
>GO

>ACCESS sales-mast LINK TO RECORD(0) OF *myfile
>DEFINE sales-pct = (sales-dollars/total-sales) * 100
>REPORT &
>  sales-name &
>  sales-dollars &
>  sales-pct
>GO
```

CASE 3: Produce a report of all sales transactions which occurred during a particular day. This data is stored in an IMAGE data set which contains over 500,000 records for the past year. *NOTE: Rather than use QUIZ to serially read each of these records, use SUPRTOOL (or similiar product) to extract the ones you need to a QUIZ subfile and then produce the report from this subfile.*

```
:QUIZ
>ACCESS sales-dtl
>REPORT SUMMARY all
>SET REPORT LIMIT 1
>SET SUBFILE NAME psales SIZE 2000 KEEP
>GO
>EXIT

:SUPRTOOL
>BASE mybase,5,password
>GET sales-dtl
>IF trans-date = 910501
>OUTPUT psales,ERASE
>EXIT

:QUIZ
>ACCESS *psales
>REPORT ...
```

Paper #6110 - How To Survive As A Small Shop Manager

John T. Monaghan, CDP
Rudolph and Sletten, Inc.
989 East Hillsdale Blve, Suite 100
Foster City, CA 94404-0637
(415) 377-1825

Introduction

As the economy expands and contracts, small companies are being forces to rely more on computers for faster analysis of operations, both cost and performance. As smaller companies place more emphasis on these systems, they also place more emphasis on the people charged with the care of these systems, the Data Processing Staff. The subject of this paper is the survival of the Data Processing Manager in these organizations.

More often than not the manager serves in a company where the company realizes the importance of the systems but doesn't always appreciate the importance of the manager. We receive the proper amount of respect due a manager on a day to day basis but are rarely recognized for our technical skills. When things go bad, we are subjected to the universal demand "it worked, now it's broke...fix it".

In this paper I've tried to assemble key points I've developed in the 11 years I've had my shop. In addition I've added some thoughts and observations made of and by other small shop managers. The goal is to provide some assistance to others who are working their way through these changing times. The intent is to provide some help in surviving as the manager of the small shop.

Before I proceed with this paper, however, I would like to call you attention to the definition of the word *survive*, according to Websters' Dictionary:

1. To remain alive or in existence.

2. To continue to function or *prosper.*

Looking into Webster's again we see that *prosper* is defined as to succeed in an enterprise or an activity.

Based on these definitions, as Data Processing Managers, we have a choice. We can either

continue operate as we do today or we can prosper.

Changes in Data Processing

During the past five years the economy has gone through rapid changes, both positive and negative. In order to deal with the economy, small companies are being forced to rely on the use of computers in their day-to-day operations. Historically, companies computerized various aspects of their administration and operations to reduce overhead costs and establish an efficient environment for the business. Within the last five years, these companies have been under increased pressure from both the government and other businesses to use the computers for more of their administrative tasks.

For example, the government now requires wage and accounts payable information (W2 and 1099 forms) to be submitted to the IRS on magnetic media. Manufacturers will soon require that suppliers submit payables data through EDI (Electronic Data Interchange). Large companies are paying bills through bank wire transfers. Companies are communicating to each other through electronic mail services.

These services and requirements are all relatively new to our industry, Data Processing. Ten years ago we were concerned with developing systems for general ledgers, payables and other financial subsystems. For those in the manufacturing industry, there was the introduction of computerized MRP systems. Other industries were quietly working through implementations of their own. It wasn't until the mid 80's that there were industry wide requirements, such as those listed above, to be dealt with.

As companies begin to accommodate these new requirements, Data Processing goes through their own changes. For too many years we were operating in a reactive mode, responding to management's needs. Now we are trying to learn these new technologies quickly, trying to become fairly well acquainted with several new projects while trying to anticipate management's needs.

My own experiences mirror the industry trends. In the mid 80's I was hard at work on a system conversion. Management was told that the new system would carry the load for five years, or so, before we needed to take a look at anything new. At four years I was looking elsewhere. It wasn't just because the existing system was overwhelmed; the requirements had changed. In 1985 our business was best served by a strong centralized system. In 1989 we saw the need for a decentralized system that more closely matched the changing structure of our business. The job of getting the company there falls to Data Processing.

The Small Shop

This phrase is used extensively in the trade press, usually when the writer wants to point a

failing of small organizations. If you were to take a long look at the definitions that appear in the trades, you would come away thinking that the small shops are mostly staffed with 'wanna-be' DPers, operating without the important things in life, such as IBM *iron* and SNA. My own experience has shown me that this shop bigotry is quite real and is a very strong feeling that pervades the computer rooms throughout the fortune 500. I used to work in one such facility.

What is a small shop, if it's so bad? What sets it apart from the large facilities? Is there a mathematical equation that can be applied to certain values found in a shop that would lead to its classification as a small or large shop?

Since there's no such formula, I'll give you my definition:

> A shop is small when the budget for staff, supplies support contracts is less than $750,000. That's where I draw the line. I don't take into consideration the amount of money spent on hardware purchases and leases in determining size. I don't consider the amount of money spent on software purchases either. These are usually asset purchases and are part of the corporations' overall fixed asset program. I look at the amount of money spent in staff salaries, supplies and support contracts when I am trying to define size.

Granted, there are always exceptions to this definition but for the purposes of this paper, we'll use it.

Staff Size and Organization

According to most business publications and textbooks, the shop should be organized as seen in the chart 1. (These charts are found at the end of the text.) This organization provides for a limited span of control and individuals have specific command paths. In short the people in the group should know who they work for and what they have to do.

For several years I used this structure and organized the departments' activities around the chart. I developed the job descriptions around this structure. It worked quite well for years, until we introduced personal computers and decentralized accounting into the picture. The system worked as long as the staff had very specific tasks to perform and they were able to keep up with the workload. Chart 2 shows how the work was assigned. Notice that the work assigned to the Production side was specifically in support of the day-to-day needs of the computers. These tasks, ranging from data entry to performing backup, are all tasks that are normally associated with operators. The Programming group dealt only with application programming.

In 1986 I started to bring personal computers into the environment. At first they were more

novelty than anything else, as far as my staff was concerned. We were responsible for assembling them, loading software, fixing broken systems and just about anything else that you could associate with them. At first I took on the responsibility of trouble shooting. With the original 10 systems there wasn't much work involved.

In 1989 the original 10 grew to 130 systems. We were beginning to use them for more than Lotus and WordPerfect. New systems were being developed that placed the PC into the environment that was reserved for the minicomputer. Our users were spread around the state, utilizing the PC for remote access to our minis. Our job descriptions and organizational structure were not able to handle the new demands made on our department.

In the past I had been assigning specific tasks to individuals. I went about task descriptions following steps that I had learned in years of management classes on the job and at college. You take the number of bodies you have and divide up the tasks among them. You can also do the reverse; take the tasks and divide them up to fit a number of people. The procedure is supposed to be an exercise in objectivity.

The objective method works best when you need to survive on a daily basis. We were just getting by, no progress was being made, no careers were advancing. Everyone was getting stressed out and tempers were short. Looking back on it I would say that we all knew that we were not content with the old way, we wanted to move on.

The new question was not in how you organized the staff to cover the tasks required but how could you design the tasks so that people could cover them and still have the opportunity to advance.

Going for a non-traditional approach, I divided the department into very general functions and then defined those general areas into specific tasks which I could group together so that related tasks could be given to a single person. Look at chart 3 to see the new organizational chart. It doesn't look like the chart seen in management texts, but it works just fine.

The point I want to make here is that you should not try to fit your group, no matter what the size is, into clearly defined positions. I've found that you need to be flexible in the small shop. As requirements change and user's needs vary, you'll need to have a staff that's flexible enough to meet those demands and still be able to carry out their primary tasks.

In developing a flexible staff I realized that there is a need for cross training. In a small shop the most demanding time isn't necessarily when user needs peak or systems go down. More difficulties are encountered when people go on vacation. In a large shop there is usually more than one person responsible for a given task. When the individual that is

primarily responsible goes on vacation, there is a second person there to cover things. In the small shop there just isn't enough people to do that.

What I have done is to allow an individual to work with another person, thereby learning a little of the others' task. For example, my PC technician is able to install and troubleshoot PC software. This solves a number of problems. I have built in backup for vacations and other absences. Most important of all, my staff is able to take care of the current problems while working on new projects and learning new technologies. With a staff whose size hasn't changed in five years I've been able to *prosper.*

The staff is handling most of the day-to-day problems and I am able to spend my time on administration and planning. In most cases, unless a problem is major, I don't hear about it until after its been solved.

One of the questions I've asked other managers deals with the dependance management has placed on the manager. In a number of shops the manager spends quite a bit of time working with management on systems analysis that could be better handled by someone in the programming group. It doesn't appear that management puts a lot of trust into the DP staff, once they get past the manager. Other managers have commented that both the DP staff and the users seems to flounder when the manager is not around. Problems that do come up can usually be handled by the staff without the managers' help, but the manager seems to be used as a crutch.

To handle this problem I've placed more emphasis on the development of the senior members of my staff. Not only can they make decisions in my absence, they can make those same decisions when I'm there. They are responsible for a lot that occurs in the department and they have some control over how it occurs.

Upper management, on the other hand, presents a different problem. I feel that it is in the best interests of the department for the manager to be the link between management and the DP staff. I have found that most planning discussions that involved my staff and upper management have broken down into one-sided affairs with management dictating 'the view'. I have no problem with the occasional meeting where questions from both sides are voiced about direction, performance and the future. They help keep the staff focused on the plan. For specific discussions between my staff and management, I am there with my staff member, to be sure that they are participants in the discussions.

It is the managers' job to keep management informed about DP's plans. As system requests are submitted for approval, the manager needs to be able to keep management appraised of the schedule and the impact of changes upon it. The manager needs to be very familiar with managements' views on DP and the future of the company in general. He can

communicate this information to his staff.

The manager should be a buffer between management and his staff. He should do what he can to see that the staff can perform their tasks with minimal interruption. The staff should also be given the authority and responsibility to accomplish the job. Most importantly, the manager shouldn't be doing everything himself.

Time

Time is the scarcest resource the small shop manager has to work with. I could bore you with several cliches about the lack of time but we all have experienced the problem. In the small shop, where your staff resources are very limited and there isn't enough manpower available to cover everything in a timely manner.

The key to keeping your staff on track lies with the manager. I know this to be true from my own experiences. Several years ago I decided to keep a log of how much time I spent working up and revising project schedules. In an average month I was spending just over 25 hours. Add to this the 16 hours that each staff member was tying up working on schedules and you begin to see that this is a fairly expensive proposition.

There had to be a more efficient way to handle this and I experimented with a few solutions before I settled on the current method. I tried computerized scheduling systems and found that the time it took to set one project into the system was often greater than the time the task actually took to accomplish. I tried timesheets with no real success. The problem there was that the staffer filled the sheet out at the end of the week. This was great if you worked on a large project but most of the work we were doing could be classified as short-lived projects, maybe two to four hours in length.

After many attempts I realized the problem: I was trying to schedule everything when all I really needed to do was keep track of what I was doing. The schedule would take care of itself.

Let me define the preceding statement a little more before I share the solution. As the manager, we are constantly being pulled from one task to the next, and this 'pulling' isn't always coming from a source that we can control. Top management, for instance, doesn't care that we are working on system configurations when they want an explanation for recent expenses. Other interruptions, such as vendor visits and user meetings, could be scheduled for specific dates and times. If I kept track of these commitments and laid out my tasks around them, I would end up with a schedule.

This concept wouldn't work for programmers, they shouldn't have the number if scheduled interruptions the manager has. Also, they should have a small number of concurrent tasks

to fit into their day.

Now that I had the system, I had to find tn implementation tool. Once again, hitting the 'self-help' books and asking around, I couldn't find a consensus answer. In Jeffery Mayers' book, *IF YOU HAVEN'T GOT THE TIME TO DO IT RIGHT, WHEN WILL YOU FIND THE TIME TO DO IT OVER?*, the author suggests two tools:

1. A Calendar. Only 1 calendar is needed, add more and you're back where you started.

2. An appointment book. The calendar shows a day at a time, the appointment book shows your day down to the minute and they must agree with each other.

I've used this system for a while now and have found that it works quite well. I keep the calendar and pass information to my administrative assistant as necessary. The key is to keep control of the calendar myself. Chart 4 shows both the monthly calendar and a few pages from the appointment book.

A key point that I'll stress is that you take this appointment book with you, as you move through the day. It's easy to commit to a date and then forget to write it down when you get back to the desk. It's easier to take the book out during a meeting and check your schedule before committing.

I've found that getting control of my time has given me more time to do serious work. This was accomplished using only the two tools and didn't negatively impact anyone else on the staff. On the positive side, I'm not as stressed when problems arise and I have to drop everything else to handle them.

Outsourcing

A new word started to appear in the last two years, *outsourcing*. Computerworld has been running articles on outsourcing for a year now. The CIO magazine IS Manager of the year is the DP manager at Eastman Kodak who significantly reduced IS costs by outsourcing almost all of the corporate IS functions and reducing the size of her organization considerably. The word doesn't appear in Websters yet so I'll improvise.

The term *outsourcing* refers to the practice of contracting with vendors to take care of tasks formerly assigned to the DP staff. The concept is old, going back to the earliest days of our industry, when clients hired IBM do run their systems.

There are many benefits in outsourcing in the small shop environment, using vendors this way. Without hiring more staff you can maintain your programs using contract programmers

or consultants. There are many sites in this area that use temporary or contract operators for backup processing. A more extreme example would be Eastman Kodak, where vendors handle all personal computer purchases including software, hardware and service contracts.

By using the vendors you accomplish a few things.

1. A task that would have tied one of your staff up is given to a vendor who is capable and qualified to handling it.

2. You can probably find a vendor who is anxious to work with you and develop a long term relationship with your company. Money does breed a certain loyalty.

3. A vendor is probably move cost efficient than hiring someone new.

4. When problems arise, you have someone to call on for help.

In the small shop you need to take advantage out outside sources for many activities that are time consuming. I've been using vendors for a few years now and find that the benefits far outweigh the costs. The following are a few of the activities I've *outsourced*:

1. COMPUTER SUPPLIES. After some searching I was able to find a vendor that was able to supply me with the materials I required and handled ordering. He comes into the office every two weeks and takes a physical inventory of all ribbons, diskettes, paper, etc. that are used and reorders as necessary. We meet every quarter to go over consumption and costs, making changes to stock levels and brands.

2. WIRING. With over 70 personal computers in our building my staff couldn't keep up with wiring problems. Once again I was able to find a vendor to handle this for me. His firm will make any wiring changes I need within 24 hours of my phone call.

3. PC SOFTWARE. With the number of software dealers in the SOUTH BAY, it wasn't too difficult to find a few who offered to take on all our purchasing. My PC support specialist has one number to call, they handle everything, including upgrades.

4. PC HARDWARE. Once again, I have a dealer who handles this for me. I call with my needs, they respond.

5. PC REPAIRS. Here I've chosen to go with Hewlett Packard. All my computer maintenance is handled by HP. Yes, this may cost a little more but the end result is that they take care of it and I don't lose any sleep. The quality more than makes up for the added cost.

Let's look at a few of these items in detail. The first item, using vendors to handle my supplies has saved me approximately $3,600 per year. This is time that would normally be spent by someone on the staff taking care of the ordering, inventory and general cleanup. In addition to saving me a little money, the vendor keeps an eye out for bargains in our standard supply items and is able to locate special items.

When we install a PC, both hardware and software, we used to spend an average of 5 hours time loading and testing. By having the vendor handle this I've saved about $3,000 last year.

I have similar savings in most of these areas and haven't had any problems with these relationships. While I benefit from these relationships, I have to do a little extra work keeping an eye on them. I monitor their performance using several tools including their invoices, service call logs and follow-up visits to users.

There have been a few questions raised about my *outsourcing* practices, primarily from management. They want to be sure that I'm not leading the company down a primrose path by relying on outsiders. I don't believe this to be the case. There are quarterly reviews of activity between us (the vendors and my staff) and we have implemented strict rules covering the acceptance of gifts. These vendors are also aware that I review the invoices and look at prices offered by other sources.

In the long run, I've been very satisfied with the this practice. I haven't any plans to expand on it at this time but am always open to discussion.

Cost Control and the Budget

Personal Computers have made a major impact in organizations. The cost of the systems has been dropping steadily over the past few years. Companies have been acquiring personal computers at a fast rate. The cost of these systems is usually recorded as a fixed asset. The DP budget isn't charged for the initial purchase. In DP, we do our best to be sure that management is aware of the total cost of the package. The purchase decision is left to management. The same goes for special supplies and software, they are charged against the department who require these items.

Data Processing becomes involved after the system is purchased and installed. These costs

involve repairs, special boards or software used when the PC is connected to the mini-computer. I also see many of cases where DP absorbs costs for enhancements that will benefit users in other areas.

Capturing these costs and charging them back isn't always easy in the small shop. We are not quite large enough to support a full blown charge-back system but we may be looking at $100,000 a year that should be charged out. I have worked at this for a number of years, trying to make the users bear part of the costs and management has supported me in this effort.

To do this I implemented, with the assistance of my staff, a manual record keeping system. The heart of the system is a form that is used to record all services performed for any user on any system. At the end of the month we go through these forms and prepare the necessary paper for accounting, who processes the necessary journal entries to transfer costs from DP to the user departments.

In an average month we are able to redistribute $6,000 in supply and software costs. While we still absorb some costs, the users carry a major portion of the load.

A sample of our service call form is found in the attachments.

In addition to the service call form, I have implemented a procedure that requires the user to get approval for all software that isn't considered 'standard'. This approval form is attached to the invoice when received and the cost is charged directly to the users department.

Another cost category that we have to deal with is the cost of staff working on service calls. I don't charge time out to 'overhead' departments as these other departments have no control over the service call costs. They are only charged for materials not covered under service contracts. I prefer to have all the staff time charged to the DP department as it is easier to manage the cost that way. If, however, I have to use a vendor to handle a problem, their invoice is charged directly to the other department.

The annual budget is one area where I make several assumptions when projecting costs. I can assume that I'll have the same staff next year as I have now and they will be making roughly the same salary. I can also assume that some costs won't rise, such as telephones, allocated lease costs and other fixed costs. I do most of the guessing on supplies, service contracts and other miscellaneous expenses.

Service contracts can and will change several times over the year and need to be watched carefully. I start the budget process by taking a look at contracts that are in force now and use their current charges to base the original budget on. Quarterly I revise these numbers

to match revision in contracts.

As for supplies, some I can put a fair estimate on, such as ribbons, paper, diskettes and tapes. The quantities I use next year will be at the same rate as quantities used in the last three months of the current quarter. I'll start with these numbers and revise quarterly.

Miscellaneous expenses are an area that I have no control over and I don't enter an original budget form them. As these expanses come up I revise the budget accordingly. I also keep the controller appraised of any major purchases that I'll be making in this category.

There are other costs that I plan for on an annual basis. These include professional association dues, magazine subscriptions and travel costs for Interex meetings. I always budget for a few conferences and classes each year, making it easier when obtaining approval.

Another segment of the cost control process is staff involvement. Everyone needs to be involved in the process, from the manager on down to each member. Cost containment isn't a subject that gets dusted off once or twice a year, brought out for those occasions when you feel you need to crack down. While staff members may not be interested in saving the company money for the sake of the company, they can easily be encouraged to save when they understand that the money they save has a direct impact on the profit sharing plan.

Staff Development and Retention

In a large shop, members of your staff have a good chance at being promoted through the ranks, as they grow as programmers, operators or support specialists. In the small shop, there isn't a lot of places to go. You may only have one analyst slot and two qualified people and you'll have to make the decision on who gets the job and who doesn't.

I've dealt with this problem for a number of years. My staff is quite small, two programmers, one technician, one operator and one administrative assistant. As you can see, there's not a lot of room for growth anywhere. I know it's important to develop people for new positions so that they can advance in their personal careers; just as it is important to develop people for the needs of the company. The difficult part is in deterring whose career closely matched the companies needs. When you think you've got it figured out, the company wants to implement something new and someone has to learn it.

Part of career development is in finding challenges for people. They may be very happy with their job and you've got to change the job. How you introduce that change has a lot to do with success in implementing the change. I try to make the change a challenge. I want the programmer or operator to want to take on the change, even though it's not quite what they wanted. The trick is in finding out who can do the job if pushed in the right

direction.

As for retention, people will want to leave when they're tired, bored or not happy with the way things are done. It is our job, as manager, to see that they don't become tired or bored. We have to communicate with them frequently to be sure we know what makes them unhappy and what we can do to make things better. Sooner or later they will want to leave and there's not a lot we can do about it.

I've seen mistakes made in this area and I've made a few mistakes myself. One of the biggest is to give an employee a raise just to keep them. Once an employee knows how to open the wallet they'll do it again. I now let them go when they give me the big threat, "more money or I'm gone."

Another mistake is to make changes in job descriptions in order to pacify someone. Quite often it is an easy way to keep someone for a while but in the long run it's dangerous to make the changes. In a small shop we have a limited number of people and tasks. We try to match people to tasks that are related to their education, experience and preferences. When you've got a staffer who wants to take this carefully thought out plan and manipulate it, you'll loose the organization. It's just not worth it.

Finally, the most difficult task in staff development and retention is letting someone go. There are always going to be people who can't perform at the level you desire. They may not be interested in the job and your attempts to create enthusiasm haven't gone anywhere. It's time to go.

Hopefully they'll have seem the writing on the wall and will make the break on their own. I've had a few situations where this has happened and we've parted on a friendly basis. Unfortunately I've also had the other kind, where things are said that aren't really meant and relationships can get ugly very fast. Years of management experience and training can't prepare you to face this, no matter how many times you go through it.

I handle the problem quietly and carefully. I've learned that terminating someone opens up a number of doors for litigation and every effort must be made, by the manager, to protect the company and himself from this. When I am getting ready terminate someone, do a few things to make the job go smoother:

1. Keep track of the problems. I use a diary to write down problems as they occur. I note what action I took at the time.

2. Go over problems with the employee. It's better to air out the differences as soon as they are noted. It isn't going to help them or yourself to wait a few weeks to bring them up.

3. It things keep going bad, I prepare a counseling statement and go over the statement with the employee, asking him to sign it. A copy of the statement is given to the employee while the original goes forward to my manager and eventually to personnel.

4. I will prepare a second statement within 60 days if the employee has not made progress in correcting the problem or begins to mess up other areas.

5. If two statements haven't got the problem solved, I go to my manager and personnel and explain the case, asking for concurrence in termination. Once I get approval, the employee is quickly released.

After any disciplinary action, such as termination, it's important to go over the problem with the rest of the staff where it concerns them. I've found it necessary to break the news to the rest of the staff myself, rather then to wait for the rumors to circulate.

I won't go into layoffs in a lot of detail, other than to leave you with two comments:

1. I believe management should take a hand in notifying the employee of the action. Frequently the DP manager has nothing to do with the selection process or the reasons for the layoff itself. These are management's actions usually taken when they are trying to reduce costs quickly.

2. I also believe the DP manager should be supportive of management's actions. While you may feel that you should buoy up spirits, think of what you say before you say it. Unless you also get laid off, your allegiance belongs to the company.

Your Growth

I've written down a lot of points that are meant to help you do your job and make it a little easier. Now I'm going to lay the bad news on you. Our position has gone through some changes int he last few years and will continue to do to. In a few years our position will be that of an administrator, one who knows very little about the bits and bytes of computing but is very good at coordinating, scheduling and presenting.

The time is coming soon when small companies will have an HP3000 system that is managed by a third party who is never on site. You'll not need programmers as the software you company runs is 'off the shelf'. Operators won't be needed as users run the reports themselves and the backup is made nightly on a cassette tape, which is changed daily by the same user who prints her own payables checks. The only technician present is the lan administrator; a technician in the sense he administrators the operation of the lan, all the

actual maintenance is done by the lan supplier.

We'll survive by making our services invaluable to the company, by making our position an important part of the daily operation of the firm. We small shop managers will survive when we let go of traditional DP structures and learn to adapt to the changing face of business.

In the meantime, we can become better at what we do now. Learn to utilize staff members more proficiently. They can perform a lot of tasks that you normally handle, and probably do some of them better as they don't have the interruptions you live with. Make better use of the resources you have in your department, including computers, software and other items. Watch costs and make an effort to control the costs where possible.

We are frequently hired in as manager and live with the realization that there isn't going to be a promotion for a lot of us. The usual path for a DP manager's promotion is through another company, when you get hired for what you've accomplished. I was brought into this job in 1980, as DP manager, and will be the DP manager until I finally leave. Although it can be disheartening at times, I live with it.

I look for the things that keep me interested in the job, such as the completion of a particularly challenging project. I also look for the users comments. It's amazing how good you feel when a user comes up and tells you that you're work had made his job better or easier. I'm not alone in this, other managers have fessed up to the same feelings. Those of us that have been doing this for a number of years will admit that this gratification is part of the payback. We wouldn't be doing this just for the money. I enjoy those times when I see people getting into their work and enjoying it a little more.

Summary

Wrapping this paper up is going to be hard. There's a lot that I haven't talked about and some items I've covered but would like to spend a little more time on. This paper is a little like the job, there's only so much time and so much to do, you do the best with the resources you've got at hand. You approach each of the tasks with the same enthusiasm and try to succeed.

Attachments

Chart 1 - Traditional Organization Chart

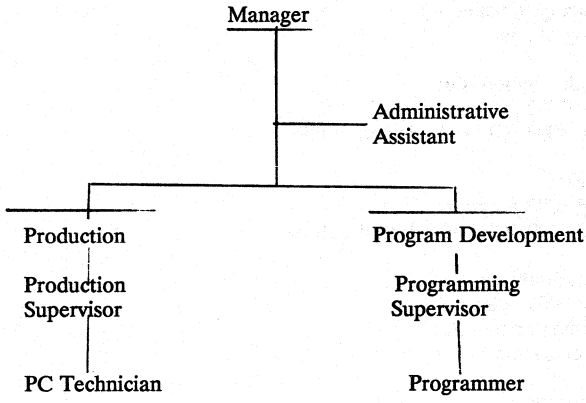


Chart 2 - Work Assignments

Manager:
- Administration
- PC Software Support
- Backup for operator
- Programmer

Programming Supervisor:
- HP 3000 programming
- HP 3000 JCL and selected utilities

Programmer:
- HP 3000 programming
- HP 3000 JCL and selected utilities

Production Supervisor:
- HP 3000 operator
- Computer supplies
- User contact

PC Technician:
- PC repairs
- HP 3000 operator

Chart 3 - Revised Organization Chart, Showing Duties

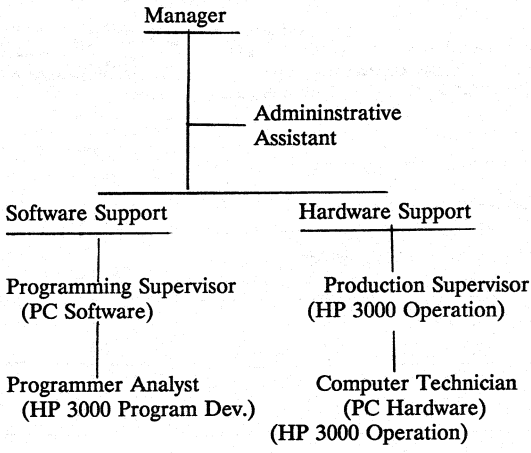
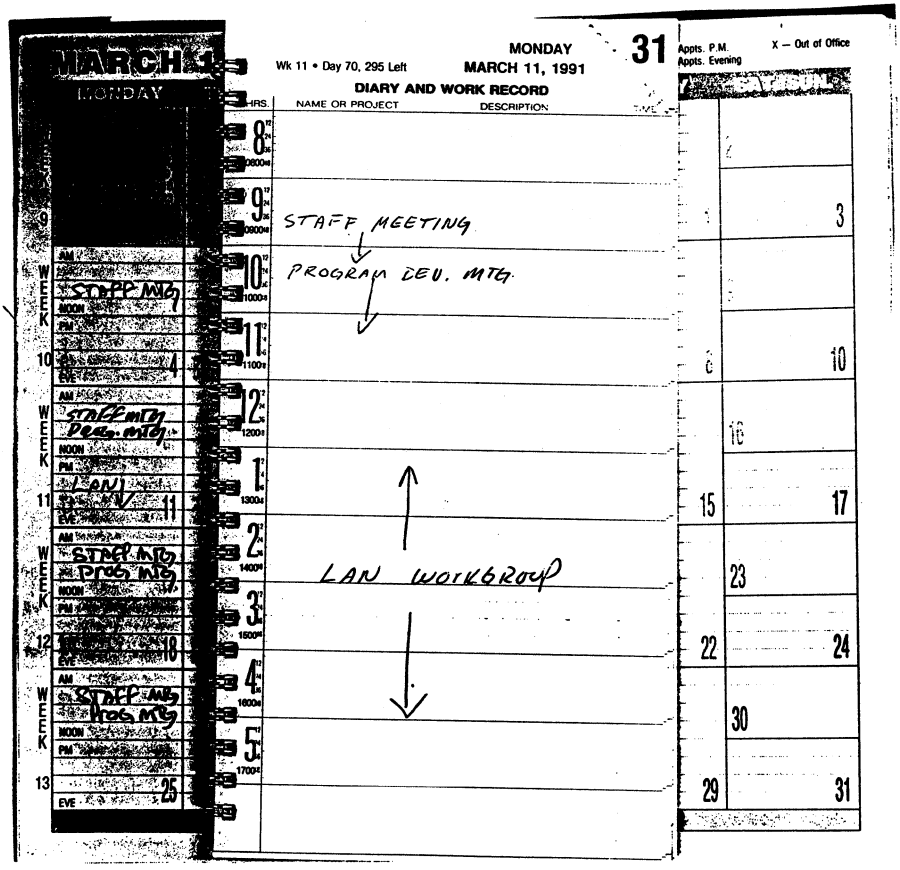


Chart 4 - Samples of Calendars



Service Order

RUDOLPH AND SLETTEN INC.

DATA PROCESSING SERVICE REQUEST

No. 00313

DATE	REQUESTED BY	PHONE NUMBER	CUSTOMER NUMBER
LOCATION		TAKEN BY	DATE PROMISED

TYPE OF REQUEST

COMPUTER ☐ HARDWARE ☐ OFFICE EQUIPMENT TELEPHONE ☐ IN-HOUSE ☐ OTHER _____

☐ SOFTWARE ☐ SUPPLIES ☐ JOB SITE

SERVICE REQUESTED

☐ INSTALL ☐ REPAIR ☐ DELIVER ☐ PICK UP ☐ ASSIST/TRAIN

☐ CHANGE ☐ COMPLAINT ☐ ESTIMATE ☐ OTHER _____

EXPLAIN:

PRODUCT	MODEL/VERSION

MAKE	SERIAL NUMBER	R & S NUMBER

SERVICEPERSON	COMPLETION DATE

	LABOR HOURS	LABOR CHARGES	MATERIAL CHARGES

6111

THE BACKUP PRIMER

JEFF ODOM AND NABIL ABOULHOSN

BAHLSEN INC.
ONE QUALITY LANE
CARY, NC 27513
(919)677-3227

Life seems to become more complex with each day. Society has sought to deal with the complexity through the use of information technology; yet most interestingly, it is the introduction of technology that has added dramatically to life's complexities. Today we depend on information technology in almost everything we do. The maintenance of our personal lives and our organization's competitive advantages are impacted by information and its nonstop high speed generation and transmission.

As information and its meaningful transfer becomes more important to virtually every facet of our daily existence, the safeguarding of this valuable resource becomes a high priority. A paradox exists in all of this as we must provide constant and rapid access to information while safeguarding it through redundant storage and security. The days of shutting the technology system down for extended periods to take backups are gone.

Today's environment demands constant vigil over the safety of the information but its very importance demands a painless process with maximum availability. Of course, unlimited budgets for such a mundane process as information redundancy are available to everyone, aren't they? Everyone in upper management has a clear understanding of information access, backups and redundancy, don't they?

Over the past three years our organization has spent significant time developing strategies and practices related to information backup and storage. Not only do we have a healthy self preservation instinct that naturally leads us to devoting time and attention to such matters but we have had a number of incidents over the past years which put our backup processes to the ultimate test - we had to use our backups.

Our company began a transition from a batch processing system revolving predominately around financials and order entry to an on-line processing and inquiry system providing solutions throughout our administrative and manufacturing operations. We moved from an 8:00 to 5:30 shop to 24 hours a day six days a week. Throughout the implementation of equipment and applications, a piece at a time, we found our backup strategies evolving; our choice was to control the evolution not let it run alone.

We intuitively have developed a process for evaluating and optimizing our backup procedures. Over the years we have reviewed several products, many which did not fit our environment. We encourage you to review any of a number of fine products on the market; here we do not intend on speaking of any particular product or its potential utilization. You should develop strategies and implement products as appropriate in your environment.

The fundamental structure for the process we follow can be described by review of the following criteria:

1. Information impact analysis

2. Application environment

3. User needs

4. Physical environment

5. Information asset deployment

In our organization today, system's administration is responsible for fine tuning adjustments to the backup procedure. Yet, on a periodic basis a committee of three sits down and analyzes the completeness of the strategies and procedures. Remember this analysis of backup must include a critical view of recovery to insure adequate maintenance. The end product is maintained in our Disaster Recovery Plan and Operations Procedures Manual.

Before we can begin, we must understand what our backup objectives are. To understand these we must understand them in light of our disaster or contingency plan, so let us first consider what contingency planning involves.

CONTINGENCY PLANNING

1. The advance planning necessary to insure continuity of the critical functions of an organization.

2. The advance arrangements to insure continuity of the critical functions of an organization.

3. Putting into place sufficient preparations in the event of a disaster. Further, designing and implementing sufficient procedures for response to the disaster and activation of preparations. Extremely important is that these preparations and procedures must be agreed to by management and participants in the disaster response team.

4. Implement procedures that will deter or reduce the risk of identifiable threats. For example, backup procedures should be designed as controls to reduce the risk of lost information.

5. Define a plan which covers and specifically determines procedures for all events of total or partial cessation of operation or destruction of the physical facility.

6. Develop the plan in such a manner that procedures, availability of equipment, personnel and physical facilities for both automated and manual critical functions are defined.

7. Develop the plan by critically analyzing impact to on-going operations as well as during plan activation.

BACKUP OBJECTIVES

1. To follow the guidelines established by our contingency plan.

2. To integrate our strategies and procedures with the varying components of a comprehensive contingency plan.

3. Optimize information integrity and security.

4. Maximize uptime.

5. Minimize the purchase of additional hardware and software. This does not imply a complete absence of evaluating additional hardware and software.

As noted above, backup strategies and procedures are controls put in place to reduce the potential and magnitude of information loss. Basically, our risk is the loss of or access to information due to some event. Backup or redundancy is one component in our information protection plan - others include site security, UPS systems, associate training, appropriate application training and so forth. Our objective here is to protect the information even if all other components of our technology system are lost.

Following is a brief description of the criteria we routinely review to determine the adequacy of backup.

1. **Information impact analysis:**

   - How critical is the specific information?
   - How critical is the timing and availability of the information?
   - How critical is the information to customer service and cash flow?

- Can the information be segregated into critical and noncritical components?
- Was the information entered in a batch control environment or on-line in a dynamic fashion?

2. **Application environment:**

- What is the purpose of the application and how critical is it?
- When is this application used?
- How is the information entered and maintained?
- Who uses the application?
- What is the size of the databases? Are there multiple in the application?
- What databases in what applications have to remain in synchronization?
- How often do application programs need to be backed up?
- Is logging used?

3. **User needs:**

- What is the scope of the positions filled by the application's primary user?
- What are the hours worked by the users?
- During the hours worked, what are the work patterns of the users?
- How critical is dynamic entry and/or access to users?

4. **Physical environment:**

- Do you have a single or multi-site operation?
- If multi-site, do you have central or decentralized processing?
- If multi-site, do you have closely located facilities (within 10 miles, 20 miles, etc.)?
- Are there sufficient and competent third party storage facilities available?
- How many systems are in place?

5. Information asset deployment:

- How many time zones are supported from the site?
- How many locations are connected to your site?
- How many and what types of devices are connected to your system?
- What are any special network considerations?

The above list is not all inclusive but will provide a sufficient basis for you to discuss backup strategies. Figure #1 briefly depicts the transition we have made over the past years. During that time, we have continually revisited our backup procedures refining them in light of the existing environment as well as short-term forecast. Figure #2 summarizes our current state. The matrix (Figure #3) is utilized to visually represent the total picture particularly as it relates to user downtime, a primary concern of ours. By using such a matrix, you can work to reduce the time slots that actually bring a user application down during a need period.

Please note, when looking at the associated figures, that this solution (backup) is delicately tailored for our environment. It is based mainly on our users' needs and the complexity of the major applications installed on our system. We are not suggesting that it is suitable for other environments but merely using it as reference in sharing our experience with other MIS professionals. We achieved our goal (being able to recover our system from backup tapes and logging files at any given time) with almost minimum resources.

We have also maintained our disaster recovery plan, not just by having regular and complete backups, but also by removing our backup media quickly from our primary site. As such, we do not feel it is appropriate to have unattended backups in our environment. Further, we store media in multiple locations within a reasonable distance of our facility (see Figure #4) to reduce risk of a single disaster taking out all of our storage sites.

We feel it is important to continually assess changing risk factors. Our world is not a static environment. It is important that we commit to continual improvement of our process. Quality backup solutions don't happen by accident, they are created.

Figure #1

THE BACKUP PRIMER

CURRENT BACKUP

A. Environment:

1. HP 3000 Series 960
2. HP 7980XC Tape Drive
3. Application Environment:
   Order Entry, On-line Sales History Inquiry, Inventory Control, Dynamic Warehouse Locating of Inventory, Purchasing, Forecasting, Production Planning, MRP, Product Costing, Financial, Budgeting and E-Mail.
4. Interfaced Applications:
   Purchasing → Financial
   Order Entry → Inventory Control
   Order Entry → Financial
5. User System Needs:
   a. Financial, Product Costing, Budgeting, Forecasting, Production Planning — 7:00am-6:00pm
   b. Order Entry — 7:00am-7:30pm
   c. Inventory Control, Purchasing, Dynamic Warehouse Locating of Inventory — 24 hrs/day

B. Current Objectives:

1. Optimize information integrity and security.
2. Maximize uptime.
3. Minimize the purchase of additional hardware and software.

C. Backup Strategy and Procedures:

1. Full backups performed on Saturday or Sunday depending on Bakery production schedule.
2. Partial daily backups divided as follows:
   a. Sales reporting and MRP databases are fully backed up at 6:00pm-no user impact.
   b. Order entry system partial backup prior to running invoice and update cycle at 7:30pm-no user impact.
   c. Partial or inventory control application and integrated databases on financial. (This step dramatically reduces downtime for the only users on the system this time of day at the warehouses.)
   d. Full partial backup except for what is found in c. above.
   e. Log files are renamed and new files built prior to backup.

Figure #2

BACKUP MATRIX

KEY:
UP & DOWN
DN/UP – DOWN AT PROCESS BEGINNING
AV – AVAILABLE NO USERS WORKING
N/A – DOWN NO USERS WORKING

		CUST SERVICE	SHIPPING	ACCOUNTING	BUDGET	PURCHASING	INV. FIN. GOODS	INV. RAW MAT.	FORECASTING	PROD. SCHD.
7:00A – 6:00P		UP	UP	UP	UP	UP	UP	UP	UP	UP
6:00P – 6:30P	RPTDB & MRPDB	UP	UP	AV	AV	AV	UP	UP	DOWN	DOWN
6:30P – 7:30P		UP	UP	AV	AV	AV	UP	UP	AV	AV
7:30P – 8:30P	PICK RUN	DN/UP	UP	AV	AV	AV	UP	UP	AV	AV
8:30P – 11:30P	OE BKUP, INVOICES & RPTDB MNT.	DOWN	DOWN	AV	AV	AV	UP	UP	AV	AV
11:30P – 12:00	INVENTORY REP. INITIAL DATA CAPTURE	AV	DOWN	AV	AV	DN/UP	DOWN	DOWN	AV	AV
12:00 – 2:00A		AV	UP	AV	AV	AV	UP	UP	AV	AV
2:00A – 2:30A	INV. CTRL. PARTIAL	N/A	DOWN	N/A	N/A	N/A	DOWN	DOWN	N/A	N/A
2:30A – 4:00A	PARTIAL	N/A	DOWN	N/A	N/A	N/A	UP	UP	N/A	N/A
4:00A – 7:00A		AV	UP	AV	AV	AV	UP	UP	AV	AV

Figure #3

MEDIA PHYSICAL STORAGE

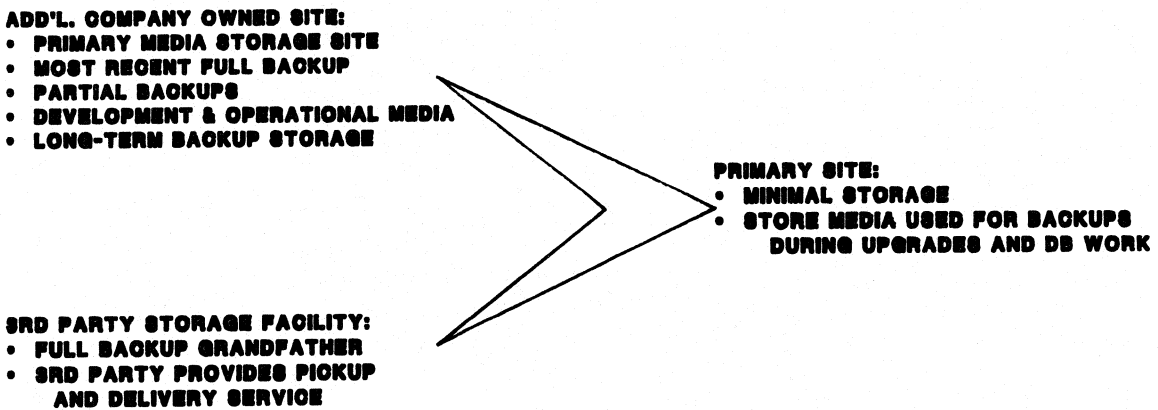


Figure #4

TITLE: System Management - Baptismal By Fire

AUTHOR: Susan Waller

Cognos

2301 E. Lamar
Suite 250

Arlington, TX 76006

(817) 649-1944

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO. 6112

Copying Files in MPE/XL

Joseph Feiner

Hewlett-Packard
19447 Pruneridge Avenue
Cupertino, CA 95014

Introduction

The copying of files is a basic service provided by any operating system. There are several methods to do this in the MPE/XL operating system. This paper will show several methods of file copying through a series of examples.

Copy

`Copy` is the new file copying command being introduced in MPE/XL. It is faster than other methods and its syntax is simple and straightforward. However, without the use of a command file (discussed later in this paper), it doesn't copy groups of files, and can only be used to copy one file at a time. Furthermore, it cannot be used to copy over a network and has limited support for file equations.

Here is perhaps the most basic example of its use:

```
Copy Wombat,Squirrel
```

Copies the file Wombat to the file Squirrel within the same group and account.

```
Copy Wombat,Squirrel.Rodent
```

Copies the Wombat file into the Rodent group.

```
Copy Wombat,Squirrel.Rodent.Animal
```

is not allowed unless you are in the Animal account since in MPE you are not allowed to copy a file into a different account than the one you have logged on to.

However,

```
Copy Wombat.Rodent.Animal,Squirrel
```

is allowed, even if you aren't logged into the Animal account as long as you have "read" access to the Wombat file.

Fcopy

In MPE V, the primary method for copying files remains the `Fcopy` subsystem. The `Fcopy` subsystem is still available in MPE/XL, but most users find it less convenient than the `Copy` command for simple file copying.

Here are the same examples using `Fcopy` commands:

```
FCOPY FROM=Wombat;TO=Squirrel;New
```

and

```
FCOPY FROM=Wombat;TO=Squirrel.Rodent;New
```

and

```
FCOPY FROM=Wombat;TO=Squirrel.Rodent.Animal;New
```

wouldn't be allowed unless Animal was your logon directory since in MPE copying files to another account isn't permitted.

Like the `Copy` command, `Fcopy` does not support wildcards. See Appendix-1 for a list of features provided by `Fcopy` that are not provided by the `Copy` command. These features include: translating a file between ASCII and EBCDIC, writing a file to tape, copying a KSAM file, and selecting specific records of a file.

Dscopy

In both MPE V and MPE/XL, the `Dscopy` subsystem can be used to copy files. It is the method of choice for copying files over a network since it was designed with this purpose in mind. It is sometimes faster than `Fcopy` (but generally not as fast as the MPE/XL `Copy` command), and will work on groups of files. Like `FCOPY`, it supports a large number of features. See Appendix-2 for more details.

Using the Print Command to Copy

The MPE/XL `Print` command can also be used for file copying. Its internal structure is most like `Fcopy` since it works on a record by record basis and supports most types of file equations.

Here are some examples of using the `Print` command to copy a file:

Prior to MPE/XL 2.1:

```
File Goldfish,NEW
Print Herring,*Goldfish
```

As of release 2.1 of MPE/XL you can use the following syntax:

```
Print Herring > Goldfish
```

The above example makes use of file redirection provided starting in version 2.1 of the MPE/XL operating system. By default, Goldfish would be a temporary file of variable length with 256 byte records.

One limitation of copying with the `Print` command should be noted however; `Print` is limited to copies with record sizes of 270 bytes or less.

Command file to Copy groups of files

Command files are also new for MPE/XL. Since `Copy` only works on a file by file basis, here is a command file that will work over filesets:

```
parm fromset="*",togroup=".!HPHGROUP"
comment
comment  executes the copy command over a wildcard set
comment  to the specified group.
comment
comment  Example:  COPYGRP @.GROUP1,.GROUP2
comment     would copy all the files in GROUP1 and put
comment     them into GROUP2.
comment
if "!fromset"="*" then
   echo usage: copygrp fromset [.togroup]
   return
endif
file garbm;rec=-80;msg
errclear
```

```
listfile !fromset,qualify >*garbm
continue
if cierror = 0 then
  while finfo('garbm','eof') <> 0 do
    input _copygrp_aname <garbm
    setvar aname rtrim(_copygrp_aname)
    copy !_copygrp_aname, !togroup
  endwhile
endif
purge garbm,temp
deletevar _copygrp_@
reset garbm
```

It would be invoked by:

```
COPYGRP @.Rodent,.Mice
```

In Dscopy this could be done with the commands:

```
DSCOPY @.Rodent to @.Mice
```

but this generally isn't as fast and would require that the network services be started on your system. However, this is the easiest method to copy groups of files over the network.

Fcopy doesn't directly provide services for multiple file copying, but you can use file equations to copy files over a network.

Here is an example:

```
file xxx=gerbil:system1
fcopy from=*xxx;to=hamster;new
```

would copy the Gerbil file that is on the remote machine, System1 to the local machine and call the new file, Hamster.

SOME OPTIONS

The `Copy` command has several options that are worth noting. One important option is the "YES," option.

If you do the following command from a session:

```
COPY Goldfish, Minnow
```

When the Minnow file already exists, you will be warned:

```
PURGE OLD MINNOW.FISH.ANIMAL?
```

Say that you would always like the new copy to replace any old version.

Then use the command:

```
COPY Goldfish, Minnow;YES
```

You won't be prompted to replace Minnow if it previously existed. The "Yes" option is the default for copying in a batch job.

Head and Tail using the Print Command

Say that you want to emulate the UNIX* commands "head" and "tail," the `Print` command can be modified to do the job. The "head" and "tail" command is used to print the first and last few lines of an ASCII file.

Command file for Head:

```
parm file,listlast=22,listfile=$stdlist
print !file;end=!listlast;out=!listfile
```

Command file for Tail:

```
parm file,listlast=22,listfile=$stdlist
print !file;start=-!listlast;out=!listfile
```

Here is a simple invocation of the above two command files:

```
Head Gerbil
Tail Hamster,12
```

The above examples show the use of START and END in a command file. The first example would print out the default number of lines to the screen (set to 22), and the last example overrides the default and prints out the last twelve lines of the Hamster file.

And by combining techniques used in file redirection you can use these commands:

```
Head Gerbil > Mouse
Tail Hamster,12  > Rat
```

Which would create a temporary file, Mouse, containing the first 22 lines of Gerbil and a file Rat, containing the last twelve lines of Hamster.

Summary of When to use Which Command

The following table shows which command is appropriate under what general circumstances:

When to use which Command?

	COPY	DSCOPY	FCOPY	PRINT
Network	No	Yes	Yes	Yes
Wildcards	No	Yes	No	No
Special Functions	No	Yes	Yes	No
Can Break/Resume	No	No	Yes	No
OK in Break	Yes	No	No	Yes

Performance

Here is a table that shows the relative performance some of the commands we have discussed so far:

Relative Copy Performance
386k sector file
(in seconds of elapsed time)

	2.05	2.2	3.0
COPY	533	613	357
FCOPY	381	395	410
DSCOPY	384	448	451

70k sector file
(in seconds of elapsed time)

	2.05	2.2	3.0
COPY	96	109	72
FCOPY	130	135	144
DSCOPY	69	85	90
PRINT	66	71	74

In this test, version 2.2 was patched with DXF4, a fix for a performance problem that seriously hurt performance for large file copies. The patch is available for general release. The comparison was run on a Series 930 with 24 megabytes of memory.

For the 386k sector file, `Copy` is the fastest way to copy in 3.0, and `Fcopy` is the fastest way to copy prior to 3.0. For the 70k sector file, `Copy` is the fastest way to copy the file in 3.0, and `Print` is the fastest method prior to 3.0. `Print` isn't shown as a method for the 386k sector file, since the record size for the file was larger than the 270 byte limit of the `Print` command.

How did Copy Performance Improve?

How did the lab improve `Copy` performance for version 3.0 of the operating system? Subsystems were invoked with explicit, time-saving options rather than their default counterparts. Functions were gathered together and done all at once yielding additional savings. For example, file initialization with the "fill" character was unneeded since the entire space would be occupied by the copy of the new file. File allocation was done all at once, in advance. Prefetching memory was done explicitly rather than relying on the less precise implicit method that happens by default. This also had the advantage of releasing memory resources as the memory was no longer needed. Files were opened in "raw mode", and operations were performed by ourselves rather than incurring the penalty of making explicit file system service calls.

The Virtual Space Manager's "contiguous block method" was employed allowing the file to be viewed in larger chunks. This made `Copy` more efficient and had the added bonus of allowing KSAM/XL files to be copied for the first time. KSAM/XL files have a gap in between their key information area and their data area and without coordination with the Virtual Space Manager, the "hole" would be copied resulting in a huge waste of cpu time and disc space. This technique has already been used with success in TurboStore.

What has been Happening to Copy Lately?

Starting in MPE/XL 3.0 performance has been improved by around 40% for some large files. KSAM/XL files can be copied for the first time.

In 4.0 further enhancements are planned to allow increased use of system defined special files such as $oldpass, $newpass, $null, $stdlist and $stdin; furthermore, files will be allowed to change their filecode via a new file equation in their destination.

BIBLIOGRAPHY

Hewlett-Packard, FCOPY Reference Manual, HP Part No. 32212-90003, Printed April 1990, Second Edition.

Hewlett-Packard, MPE XL Commands Reference Manual, HP Part No. 32650-90003, April 1990, Third Edition.

Hewlett-Packard, MPE XL Quick Reference Guide, HP Part No. 32650-90032, April 1989, Second Edition.

Hewlett-Packard, Using NS3000/XL Network Services, HP Part No. 36920-61000, August 1990, Second Edition.

APPENDIX-1

FCOPY

Use FCOPY to copy and translate files. You identify the input file and output file. You can request one or more optional functions, such as converting data, copying files from other systems, appending files, extracting subsets of files, or displaying binary files in ASCII format.

```
     FCOPY FROM=filename;TO=filename[;options]

Syntax of FCOPY Functions

                [;CCTL  ]
[;NOUSERLABELS][;NOCCTL][;NEW]


[ {CLEAR}[;HEX  ]                          ]
[;{KANA }[;OCTAL][;NORECNUM][;TITLE=title]]
[        [;HEXO ]                          ]

[      [;HEX  ]                                          ]
[;CHAR [;OCTAL][;NORECNUM][;TITLE=title][;LANG=language]]
[      [;HEXO ]                                          ]

[ {HEX  } [;CHAR ]                           ]
[;{OCTAL} [;CLEAR] [;NORECNUM] [;TITLE=title]]
[ {HEXO } [;KANA ]                           ]

[;DEBLOCK=logical-record-length]


[ {EBCDICIN }[ {field                 }                              ]]
[;{EBCDICOUT}[={{(field[;field[;...]] )} [,EXCLUDE] [;LANG=language]]]


[ {BCDICIN  }                                        ]
[ {BCDICOUT } [ {field                 }           ]]
[;{EBCDIKIN } [={{(field[;field[;...]])} [,EXCLUDE]]]
[ {EBCDIKOUT}                                        ]

[       {number-of-files}]
[;FILES={ALL            }]

[;IGNERR[=number-of-errors]] [;COMPARE[=number-of-errors]]
```

Paper # 6113-10, Copying Files in MPE/XL

```
[         [{+}              ][ {+}           ]]
[;SKIPEOF=[{-}from-eofs     ][,{-}to-eofs    ]]
[         [from-file-number][,to-file-number]]

[        ["characterstring"[,column][,EXCLUDE]]]
[;SUBSET=[#patternlist#[,column][,EXCLUDE]   ]]
[        [(range[;range][;...])              ]]

[;NOKSAM] [;KEY[=character-location]]


[;UPSHIFT[;LANG=language]] [;VERIFY[=number-of-errors]]
```

From the MPE XL Quick Reference Guide and the FCOPY Reference Manual.

APPENDIX-2

DSCOPY Options Summary

APP	Appends source file to existing file specified as target file.
ASC	Causes target file to contain ASCII data.
BIN	Causes target file to contain binary data.
CHECKPT	Used (along with RESTART to recover file transfers that have aborted.
CLEAR	Removes previously specified global specifications.
COMP	Compresses contents of target file during transmission.
DIR	Causes target file to be a direct access file.
FCODE	Specifies file code needed to open source file; gives same file code to target file.
FIX	Causes target file to contain fixed length records.
FSIZE	Specifies size of target file in records.
INT	Causes files to be copied using Interchange Format.
MOVE	Causes source file to be purged after file transfer.
OVER	Causes copied file to overwrite file that is specified as target file.
QUIET	Suppresses all display output except error messages.
REP	Causes source file to replace specified target file. Previously existing file of same name as target file is purged.
RESTART	Initiates the restart of a transfer that was previously checkpointed.
RSIZE	Specifies length (in bytes) of target file records.
SDEV	Specifies disc device on which source file resides.

SEQ	Causes target to be organized to allow sequential access.
SHOW	Displays global specifications in effect.
STRIP	Removes padding from records in target file.
TDEV	Specifies disc device to which target file will be written.
VAR	Causes target file to be composed of variable length records.

From the Using NS3000/XL Network Services Manual.

APPENDIX-3

How does Copy Work?

Here is a brief tour of the MPE/XL OS with respect to the `Copy` command for those who might find it of interest.

What OS Modules are used in a Copy

`Copy` uses the services of these major modules of the OS:

- CI -- invokes the copy command.
- Scanner/Parser -- parses the command, strips out and forwards the information on the parameters to the command.
- File System -- opens the files (source and target) and returns information on various characteristics of the file.
- Memory Manager -- fetches the file into Main Memory and posts the data back to disk where it is stored.
- I/O subsystem -- responds to requests from the Memory Manager and returns completion information back to it.
- Storage Manager -- allocates space on the disk for the file.
- Virtual Space Manager -- allocates the file's virtual address, maps the file to a virtual address, and handles the translation from a virtual to a real address.

High Level Outline of the algorithm:

```
Program COPY;

Begin
  GET_COPY_PARMS;

  GET_FILE_NAMES;

  OPEN_SOURCE;

  COPY_FILES;

  CLOSE_SOURCE;

  CLOSE_TARGET;
End.
```

```
Procedure COPY_FILES;
Begin
  (assume file falls on a block boundary -- if it doesn't handle last partial
  block as a special case)

  get source and target gufd and source file virtual address

  get_contiguous_block_info from vsm on the source file
  allocate the entire target file using storage management --
    no need to fill.

  For each contiguous block
     begin
     if there is more than one block
        begin
        prefetch_nowait the first block
        for the rest of the blocks
           begin
           prefetch_nowait the ahead one block
           if current block not present prefetch_wait for it
              move source to target for block length
              increment source_addr and target_addr by the block length
           end;
     end
     else handle special case of only one block;
end;
```

THE EVOLUTION OF THE M.I.S. PROFESSIONAL
or
WHAT YOUR BOSSES SHOULD HAVE KNOWN ALL ALONG

by

JOSEPH MENDELSOHN
L & J ASSOCIATES
25422-413 Trabuco Rd., #105
Lake Forest, CA 92630 (714) 586-4692

When the computer was first invented, it was a giant calculator, allowing us to determine quantities faster and, hopefully, more accurately. The definition of a computer centered around its ability to accept data, perform arithmatic calculations on that data, storing data and instructions, and output the information calculated. In scientific and accounting applications this remains true today.

As the cost of the computer went down, the size decreased and the capacity of the computer increased, more companies felt it economically feasible to take advantage of this product. Input and output devices also evolved, from punched cards and paper tape to magnetic tape, discs, cassettes, printers, pattern recognition devices (magnetic-ink, optical character, and bar coding) and, of course, the ever popular CRT. The concept of paperless computing began. (Judging from our desks, I'm not sure it ever really caught on!)

At the same time, the development of smaller, more intricate computers encouraged the development of other concepts and methods: simulations, multiple regression, linear programming, forecasting, status checks, inventory control and, most recently, robotics and artificial intelligence. New languages developed and so did the concept of data manipulation. Data took on new appearances: names, addresses, items to be tracked, etc.. Now instead of just being a Data Processing Department, information was being processed, and "Information Systems" came about.

The languages plus the design skills we use allow consistency in programming -- the ability to divide major applications into modules that can easily be programmed (or purchased), better system program maintenance and documentation, easily debugged, and conversion from one computer to the next. From basic flowcharting, we learn program design methods: Top-down, Yourdin, and Warnier-Orr, just to mention three. These structured methodologies assist

us in defining, analyzing, organizing, and dividing the information before we start programming. Hopefully, we know from this work whether the programs we are creating will accomplish what our "user" wants.

Now I've done it!! I introduced PEOPLE into this whole process. People -- the designers of computers, the operators, programmers, analysts, administrators, software and hardware developers, in-house personnel, outside consultants and the users. As a professional recruiter, I have the pleasure of dealing with all of these people at different times, both as clients and as candidates. With our intimate conversations, I have had the opportunity to examine these people and their positions within data processing. Let's look at the evolution of the people involved in our "automated society."

Our first computer professionals were technicians. The infamous "computer nerd", "hacker", "bits & bytes" guy is still with us. I went to college with these eccentric people. Taking a program from 120 steps down to 98 steps was an achievement. Put them in a room, feed them three (maybe one) meals a day and they were happy. Sleep, social lives, current events, fashions -- all of those mundane things that to us meant "purpose", meant nothing to them. Without these people, we would have not seen such a rapid development of technology. New languages, methods of analyzing data, and development of hardware can directly be attributed to these professionals. Therefore today, in some environments, these people are still valuable.

From computer technicians, data processing professionals evolved. Now there had to be a purpose to the manipulation of numbers. Business and scientific applications of number manipulation emerged. In larger, "structured" environments, like the IBM mainframe shops, division of labor took place. Systems Analysts emerged, telling the programmers what to do. Managers, Project Leaders, Systems Analysts, and Programmer / Analysts all developed from the programmer. These people started to consider all the applications this new science could work with. In the business world the following developments took place:

The manager came from one of two basic paths. Since accounting functions were the first to be automated, the Manager of Accounting often took over the responsibilities of the Computer Department. These people knew what needed to be accomplished but knew nothing about how to do it. In other companies, the manager was someone who had been a programmer and was promoted to manager. These managers knew how to

program, but did not have experience in managing or in the accounting functions.

There developed a need for someone who had a combination of accounting and data processing skills. The Systems Analyst filled that need. This individual talked to accounting professionals, determined what needed to be done, and translated those needs into programming modules that could be handed over to the programmers.

Then accounting professionals became users, and a formal relationship began. At first, data processing was a tool of the user -- its purpose was to do what the user needed to have done. As the Data Processing Department worked with more users within the company, the demands on the D.P. department changed. Now the D.P. Manager had to start placing priorities on the work requested. It became a "company department" instead of a division of Accounting, evolving into an Information Systems department, or M.I.S. group.

Of course, as with any developing concept, the pendulum swung the other way, and some M.I.S. departments tried to dictate what could be done and what could not be done, claiming the integrity of the system was paramount in making these decisions. These departments forgot that users are customers, and unless you take care of your customers, you lose your business. Most professional M.I.S. groups are more concerned with HOW to get things done, not whether they can be done.

Depending on the size computer, the complexity of the system, the company's software needs and its budget plus a myriad of other factors, M.I.S. groups differ tremendously. To describe the optimum department within an optimum company, certain assumptions must be made:

* The company considers the M.I.S. Department part of its decision making team, not a department that only reacts to decisions being made. One of the best arguments for this concept is using the computer for forecasting and simulations -- two tools used in making decisions.

* The M.I.S. Department is a company department on a par with other profit-making departments. Although it does not generate income, properly used it ensures the profitability of the company.

* The company has considered the expansion plans for the company in the next five years, what M.I.S. changes are being anticipated, and which problems have the highest priority. Without direction, you constantly react instead of plan.

The M.I.S. Department, under the direction of its manager, must also consider its plans. Is the department fulfilling its obligation to the company by taking care of all its users using agreed upon priorities? Will new products be needed? What does your organizational chart look like? What responsibilities are not being met by your current personnel? Are these "missing areas" essential? Should current staff be handling these items? Is additional staff necessary? Most important of all, what qualities and qualifications do we need in each position within the M.I.S. Department? Let's look at the common qualities needed in each position.

The M.I.S. Director (V.P., M.I.S., D.P. Manager) has certain basic responsibilities:

* Manage staff--hire, fire, train, review, motivate
* Handle administrative duties to the satisfaction of the company
* Needs analysis -- distinguishing between need and want and determining how it will integrate itself within the company's goals
* Set departmental goals
* Software and hardware evaluation and purchase
* Software and hardware releases (updates on systems)
* Conversions if necessary
* Software development/programming
* Systems/user support

Some of these responsibilities are delegated, however the Director should be able to do anything himself/herself. (Not have to do everything, just be able to.) Skills should include technical expertise in your environment, a background in your company's product, people management skills, organizational skills, budgeting and forecasting skills, and an ability to talk in both technical and lay language.

Depending on the size of the department, there may be a Systems and Programming Manager and an Operations Manager. These positions must have expertise in their areas of responsibility. Operations Managers should be able to anything done by an Operator; programming Managers must be able to do anything that a Systems Programmer and/or Applications Programmer can do.

Systems Analysts are a large shop luxury. This individual must have an in-depth knowledge of the business application from a user's view combined with a knowledge of the structure and design of the supporting program (or proposed program). Here we see the need for the organizational skills and the ability to talk in both technical and lay language. Most shops will either require a specific design methodology or the individual must learn the department's own methods.

Programmers write the actual code for the computer. The term "programmer" is usually broken down into two positions: Applications Programmer and Systems Programmer. Here you need someone with experience in the language your department is using. Experience in the computer you use is not necessary for the Applications Programmer, but it cuts down the learning time. A lower level of communication skills is necessary, and there is no need for the analytical skills described above. If the individual is going to be promoted within the department, however, these skills are necessary and should be considered while hiring.

Operations personnel are responsible for the actual running of the computer and preparation for data input and output. The Computer Operator should have a two year Associate or Computer School degree. If you cannot offer promotion into programming within one year of hire, make sure the candidate you hire is not looking for that promotion. Otherwise you will just have a frustrated operator on your hands.

The Programmer/Analyst is a hybrid position that was created by the mini-computer marketplace in order to allow smaller departments to have all the skills a larger department has. This individual varies in years of experience depending on the departments' perceived needs. Coding, analytical, organizational, and user interface skills are all needed. It helps to have a working knowledge of the users' applications, but usually this is sacrificed for the technical expertise.

Up to this point we have reviewed history and established what could be considered an "optimum" data processing department with its proper personnel. Of course the descriptions I gave are theoretical; each department's needs differ so radically. In addition, people we deal with are never 100% of what we wish, usually they're 70 to 80% of our original requirements.

At this point in the development of hardware, software and languages we reach a relatively new development -- the introduction of "4GL's" (Fourth Generation Languages). 4GL's are designed to allow someone to program without the extensive coding required in COBOL or FORTRAN. Let's examine some of the effects from this new development.

One big change is the users' ability to work with these "simpler" languages. Users can now define their own environment instead of responding to the environment. They can extract and manipulate data and often times write their own reports. The other major change with 4GL's is the ability to quickly and easily create prototypes for the user to examine and react to. Instead of creating a formal analysis with the M.I.S. professional the user can request changes to a tangible product.

Theoretically the net result is a saving of time for the M.I.S. professional. In addition, the user is more computer "literate", accepting the computer as a valuable tool.

New questions are raised as well:

* With the user manipulating data unsupervised, is there redundancy?

* Is the structured design accepted by the M.I.S. group compromised?

* What happens to the system's overall productivity? Could a user program consume more time than one created by an experienced professional?

* Does a 4GL inherently run slower?

* Who has ownership of the data?

* Is system security compromised?

According to a number of HP professionals, productivity goes down when the 4GL environment is introduced. Not only

does a 4GL run slower, but inexperienced users often program "inefficiently" as all beginners do, slowing the overall performance of the system. In addition, a greater number of people are now on the system, taxing the system even further.

With the user claiming data ownership, manipulating the data without supervision is probable, and the integrity of the data is then compromised. In addition, it would be reasonable to expect redundancy to occur, either by two users or by a user and an M.I.S. professional.

Using COGNOS's 4GL, POWERHOUSE, systems are scenario driven. The methodology is quite simple and consistent -- so easy a user can follow it:

1 Define
2 Analyze
3 Specify
4 Document
5 Code
6 Implement

Problem analysis involves system structure and requirements. Data analysis uses a data model, leading into a prototype. After testing and changing, the system is ready to implement. Accurate requirements and good designs are still necessary, and audit control must be included.

From this example, problem analysis remains status quo. Other factors require the role of the M.I.S. professional to change as the language changes. Emphasis must be on greater monitoring of the system, including safeguards and audit functions. User training is more demanding as the users take on more. Coding is easier now, letting the language do more of the work. Instead of the formal designing done before the M.I.S. professional is building prototypes.

As the M.I.S. professional's role changes, so must the qualities and skills we look for. Emphasis must be given to finding user-oriented people. Before we just had to get information from them; now we have to get information from them, help them manipulate data, train them, enlist their cooperation in ensuring data and system integrity and still adhere to the company's priorities.

Systems people are more important. Security must be established and monitored. Safeguards must be set up, testing must be extensive, and performance time must be constantly improved.

Now it appears depth in 4GL coding skills will not be as important because a 4GL is not as intricate. Unless all coding will be done in the 4GL, programmers will need to know the 4GL as a second language rather than a primary language requiring a certain flexibility, as one must change his/her way of thinking when switching from one language to another.

Design skills must remain important. Since more than one prototype can achieve the same results, our professional must find the optimum one. This means the one which will do what the user wants, work the fastest and still fit into the existing structure.

Other qualities and skills are status quo, skills and qualities every manager should look for when hiring or reviewing employees:

* Analytical Skills
* Communication skills, both oral and written
* Organizational skills
* Reliability -- Good work habits
* Honesty and integrity -- a personal set of standards
* A personality that will integrate well with the department's personality
* Supervisory skills for a project leader or manager
* Goals that are compatible with the goals of the department
* A "business sense"
* Someone who understands M.I.S. as part of the company, not as a company revolving around the department

This description will also fit for the M.I.S. professional of the future. From the time the first computer was designed to now, people have learned new skills, accepted new challenges, and accepted all types of new experiences. People themselves have not changed radically, however. The new complexities have increased stress, but what we're looking at is more of the same. So as evolution continues, the model employee will always be the one with the good work habits and personal skills; we will always look for that person first.

THE PHASES OF QUICK

David G. Robinson
PowerSpec Intl, Inc.
403 Cross Lake Drive
Fuquay-Varina, NC 27526
(800) 334-7023

Introduction

This paper will discuss How QUICK Works ? From the initial call of a QUICK Screen through the phases of QUICK which invoke the various procedures that control the functional activities of the screen. Several of the inherent facilities that QUICK incorporates will be discussed as well, such as the Roll Back and Backout Control facilities.

Background

PowerSpec International, a member of the PowerHouse Partner's Program. provides a complete training curriculum in the family of PowerHouse products marketed by Cognos, Inc. Since 1984 we have specialized in the training of the PowerHouse software from the first time user to the application designer with our unique curriculum. We provide training in all machine platforms that support the PowerHouse productivity tools. PowerSpec has just published its 10th Anniversary Edition of the TNT, a technical newsletter which provides Tips, News, and Techniques to a host of international subscribers. PowerSpec has hosted many Mastering PowerHouse technical seminars world-wide with much success and acclaim. These seminars provide much needed technical tips and techniques in the usage and performance issues of the PowerHouse products.

The Phases of QUICK

The calling of a QUICK screen consist of extracting a copy of the original screen into memory. There the components of the screen (1) Background (Layout of the Screen) and (2) Tables , which are comprised of the Data (Foreground) and Procedural Sections of the screen, are then mapped into the Display Buffer. A special file named Application Lines can be used for mapping the Background of up to 10 QUICK Screens (24 x 80/24 x 132). The Display Buffer then sends the Screen's Background and Foreground along with the tables into the terminal buffer for display to the user. Considerable I/O can be reduced in taking advantage of extra terminal memory by stacking screens into Application Lines. This concept is called Windowing in QUICK. This will decrease the time it takes to display a screen on the terminal window once it has been mapped onto Application Lines and is called. Now that we have the screen we will discuss in detail the various phases of QUICK and the procedural language that controls the interaction between user and QUICK. In this paper, the Timing of Procedures, Procedure's Verbs, and when to conditionally test these procedures using Processing Modes and Record Buffer Status will be explained in detail.

To understand when a procedure and how a procedure is invoked is half the battle in writing procedural code. Most procedures are invoked by an entry in the ACTION field or by specifying a value in a FIELD on the screen itself. But, there are exceptions to all rules. When a QUICK screen is called, the first procedure that will always be invoked and executed if present is the INITIALIZE Procedure. This procedure might be used for security checks to restrict unauthorized usages or maybe for specialized audits or initializations. The last procedure that is executed before exiting the screen is the EXIT procedure. This could also be used in conjunction with the INITIALIZE procedure for audits or additional customization to the files and/or items within the screen. One can also write named

DESIGNER procedures to provide additional customization or edits to a screen. A named DESIGNER procedure which is invoked by specifying the name of the DESIGNER procedure in the ACTION field. could be used to allow users to call the dictionary utility QSHOW in order to display data dictionary information. The named DESIGNER procedure would then be coded to execute the program QSHOW. Normally when you type an ID number in the action field you will either call a QUICK screen, run a program, or maybe be prompted for a FIELD value. You could write you own specialized number DESIGNER procedure to circumvent this default processing. For instance, lets say that you have an ID number (ID 60) that allows any user to go into a QTP Maintenance Menu Screen. This menu might allow updates and modifications to your data structures. You could write a procedure to prompt them for a password first, and if the correct password is given, then go ahead and call the QTP Maintenance Menu Screen, else you display an error message "Unauthorized User". To do this you would write a PROCEDURE DESIGNER 60, and incorporate the above logic.

As I stated before, most procedures are invoked by an entry in the ACTION field. Entry Mode, the side of QUICK that allows the addition of new records is initiated by an "E" in the ACTION field. Entry Mode allows us to do data entry and is comprised of four phases. The first phase - the Initialization Phase - initializes all TEMPORARY items and items with the PREDISPLAY option. The items with the PREDISPLAY option will be displayed to the screen and then the record buffers are marked for initialization. The status of the record buffer at this time is considered a NEWRECORD [NR], not ALTEREDRECORD [-NR] and not DELETEDRECORD [-DR]. The screen's processing mode at this time is ENTRYMODE.

The second phase, the Entry Phase, consists of the PREENTRY, ENTRY, and APPEND Procedures. The PREENTRY Procedure, an optional procedure, will allow you to do specialized processing to certain fields within your QUICK screen. The ENTRY and APPEND procedures consist primarily of the ACCEPT verb. The APPEND Procedure is automatically generated under two circumstances (1) Primary file with an OCCURS option (2) QUICK file type of DETAIL is specified in the Data Section. The ACCEPT verb is the main verb of the Entry Phase and initiates the calling and processing of a number of other procedures, specifically the Field procedures. The Field procedures consist of the INPUT, EDIT, PROCESS, and OUTPUT procedures. They are invoked in that order and each of these procedures will intervene in the normal processing cycle of the ACCEPT verb. Once a value is entered in a FIELD on a screen, the status of the record buffer changes to an ALTEREDRECORD [AR]. There are ten steps that take place whenever an ACCEPT verb is encountered. Following are the ten steps and some of the editing that takes place during this cycle.

1) **CHECK(s)**
   Prompts and Get's Value from User
   Size Check
   Checks Data Dictionary Security
   Checks NOCHANGE/NOCORRECT Options
   Applies UPSHIFT/DOWNSHIFT Options

2) **Procedure INPUT**
   The first of the four FIELD procedures to intervene in the normal processing of QUICK. This procedure can be used for customized inquiries when no editing is required.

3) **TYPE EDIT**
   If a numeric field, then a numeric value (0-9) must be entered Date Validation Edits are applied.

4) **SPECIFY EDITS**
All Data Dictionary and Screen edits are applied to value entered. Such as value clauses and pattern edits using metacharacters.

5) **Procedure EDIT**
The second of the FIELD procedures. This procedure is used to apply application edits to the value entered.

6) **Store value into Record Buffer**
The status of the record buffer is now [NR/ AR /-DR] if in Entry Mode, or [-NR/ AR/-DR] if in Find Mode..

7) **Procedure PROCESS**
The third FIELD procedure. This procedure is used for application calculations and updating records in other files. Many assignments are specified in this procedure using the LET Verb.

8) **Retrieve the value from the Record Buffer**

9) **Procedure OUTPUT**
The last of the FIELD Procedures. The OUTPUT Procedure is used for customized displays. It is also used for Field Cleanups after Deletion of Records

10) **Specify Formats**
This is where Dictionary or Screen PICTURE(s) and DATE FORMAT(s) are applied

After the completion of the Normal Entry Sequence, all fields have had values placed in them or skipped over, then the Correction Phase takes over. At this time the screen's processing mode changes from ENTRYMODE to CORRECTMODE. The Correction Phase is bypassed when the AUTOUPDATE option is in effect. If this is the case then this Correction Phase is skipped and you go right into the Update Phase. Under most scenarios the Correction Phase is invoked thus giving the user the ability to change a value or enter in a value for field(s) skipped. To do this you specify an ID number in the ACTION field. Internal built in number DESIGNER procedures are invoked at this time. For instance if you specify an ID number to change a field's value, then an ACCEPT verb is re-invoked and the ten steps as explained previously are once again executed. If you specify an ID number that calls another screen, than a RUN verb is invoked to execute that screen. You, as the designer of this screen, always have the option to customized these number DESIGNER procedures by writing your own customized number DESIGNER procedure for that specific ID number. In the Correction Phase you could specify a "D" in the ACTION field, though normally this is not something you do in Entry Mode , but more in Find Mode. In this case , the QUICK screen will be cleared, and the status of the file's record buffers would be set to [NR/ -AR/ DR], an exception to the normal setting of Record Buffer Status, but since it is flagged as a not ALTEREDRECORD, the PUT Verb in the Update Phase if invoked will not do any updating. Normally, you use the "D" in the ACTION field to flag an existing record to be deleted after a record has been retrieved and is displayed on the screen [CHANGEMODE].

The last phase in Entry Mode is the Update Phase. This phase is invoked by specifying a "U/US/UR/UN" in the ACTION field. It is in this phase that the actual updating of our Primary, Detail, and/or Secondary files take place. The main verb of the Update Phase is the PUT verb. This verb will first interrogate the status of the file's record buffer to determine whether or not it is an ALTEREDRECORD [AR]. If it is an ALTEREDRECORD [AR] then next it will determine if the status of the record buffer is a NEWRECORD [NR] or an existing record [-NR]. In our example the status of the file's record buffer is [

NR/ AR/-DR], therefore the information from the record buffer is extracted and updated (added) to the respective file or files. The Update Phase could consist of a PREUPDATE, UPDATE, and POSTUPDATE procedures, although the UPDATE procedure is the only procedure that is required. The PREUPDATE procedure is used a lot for inter-field type editing to interrogate and compare multiple field's values and to either accept and pass on to the UPDATE procedure or to reject [ERROR] and disallow the update to take place. The Update Phase is the safest place to generate your control numbers for you specific applications. For example, the creating of new invoice numbers, client, order, or payroll numbers. There may be times though because of the dependances of your application that you may need to create your control numbers earlier in the processing cycle, such as in the Entry Phase. If this is the case you might also consider writing a BACKOUT Procedure in the scenarios where you back out of the screen before updating the primary record, but after updating the control file.

The PUT verb has its own processing cycle when encountered in the UPDATE Procedure.

1) **Evaluate NEED Option**
   This option on the FILE command is used mostly with PRIMARY to SECONDARY type relationships. The concept is called extended records. The ability to update similar records from multiple files with one update. If specified will treat the record as a changed record In Entry Mode ITEM statements will not alter the status of the file's record buffer(s).

2) **Final Values Computed**
   All ITEM commands with Final Options are Evaluated and Executed

3) **Lock File**
   Locking is Enforced for designated file

4) **Conflicting Update**
   Reread existing record and compare checksums. If comparison is ok then update takes place. Else Roll Back Processing a feature of QUICK is invoked.

5) **Update Record(s)**
   If Status of Record Buffer is ALTEREDRECORD

6) **Release of Record Level Locking**
   This is an option for Vax VMS.

7) **Update Rollback Buffers**
   The Rollback Buffers are updated and a new checksum is computed.

8) **Reset Record Buffer Status**
   Status of Record Buffer is changed to [-NR/-AR/-DR] if it was a new record being added. If it was an existing record that was modified then it is also changed from an ALTEREDRECORD [AR] to a not ALTEREDRECORD [-AR].

9) **Unlock File**
   Unlocking of file(s) take place.

10) **Audit Record Written**
    If an AUDIT File is specified in the Data Section of the screen then an Audit Record is written for the file being audited. Audit Records are after images.

11) **Back Out Balancing**

This is only invoked when a existing record has been flagged for deletion [DELETEDRECORD] and it was declared as a DELETE file. Any SUM or COUNT operations will be reversed.

If you noticed in step 4 (Conflicting Updates) that an inherent facility of QUICK, Roll Back Processing, will automatically take place in conditions where a record has been modified since it was first retrieved and read. This will automatically be applied to files that have a PUT verb inside of the Update Procedure. This feature helps maintain a greater sense of integrity among the files being processed in the QUICK screen. This is a nice feature in QUICK, but, what about the times when one must use a PUT verb outside of the UPDATE Procedure. Then again you had better consider writing a BACKOUT Procedure. The BACKOUT Procedure will always be invoked when a file's record buffer has been set to ALTEREDRECORD [AR] because of changed data and then the user fails to do an Update.

That was the flow of procedures from Entry Mode which was initially invoked by specifying an "E" in the ACTION field. It was comprised of the Initialization, Entry, Correction, and Update phases.

Now let's type a "F" in the ACTION field. This will initiate Find Mode and assumes that you are trying to retrieve existing records for display, modification or deletion. Find Mode consist of several phases, the first being the Find Mode Initialization Phase. This phase is very similar to the Initialization Phase in Entry Mode cycle except there are usually no predisplay values for the screen. The status of the record buffer is [NR/-AR/-DR] at this time. The screen's processing mode is FINDMODE.

The next phase is the PATH PHASE. This phase will determine What Data is to be retrieved and How ? A predefined item PATH is always set to some number. For instance, a Primary file has multiple retrieval Access Paths, one for each key plus one for retrieving sequentially. Therefore, if a file has two keys: 1) Customer No and 2) Last Name, then as far as QUICK is concerned there are three paths into this file. The PATH Procedure is invoked during this phase and its verbs - the REQUEST verb which prompts for a key value and the LET verb which assigns a value to the predefined item PATH - are initiated. If a key value is specified then this value is placed into the Request Buffer to be interrogated in the FIND Procedure. The processing mode is still FINDMODE. There is an optional POSTPATH Procedure that can be written for customization at this point.

A special phase will be invoked, the Selection Specification Phase, when a user enters a "S" in the ACTION field. This phase is an extension of Find Mode. This phases will allow the operator to specify retrieval by a key value, by key and/or non key values, or using special features such as patterns for searching of records. If a value or values are entered, they are placed in the Request or Selection Buffer depending on whether a key and/or non-key value(s) were specified and then the predefined item PATH is set.

The Retrieval Initialization Phase now comes into play where the screen foreground is cleared, all TEMPORARY items are initialized and the record buffers are marked for initialization. Again, the processing mode is FINDMODE, and the record buffer is still considered a NEWRECORD, not ALTEREDRECORD , and not DELETEDRECORD [NR/-AR/-DR]. No procedures are invoked during this phase.

The Data Initialization Phase retrieves the records from either the Primary, Secondary, and/or Detail Files specified in the Data Section of the screen. The FIND Procedure is the first procedure invoked here, and its verb, the GET verb, interrogates the predefined item PATH that was set in the PATH Procedure, and optionally retrieves the value or values from the Request/Selection Buffer. It executes the

GET Verb to try and retrieve the record(s) and places it into the Record Buffer At this time we are still in FINDMODE, and the status of the Record Buffer is [NR/-AR/-DR]. If a DETAIL File was specified then after a successful retrieval of a Primary File's Record, then the Procedure DETAIL FIND is invoked to extract all associated Detail Records.

The Display Data Phase is the interim step of finding the record and sending it to the terminal buffer for display. While in this phase the screen's processing mode is FINDMODE. If a SECONDARY or DETAIL File has been specified then the retrieval of these records are always based on the successful retrieval of the PRIMARY Record.

Once the record or records have been retrieved, you are considered to be in the Change Phase. This is very similar to the Correction Phase on the entry side of the coin in QUICK. The screen's processing mode now changes to CHANGEMODE and the status of the record buffer is set to not NEWRECORD, not ALTEREDRECORD, and not DELETEDRECORD [-NR/-AR/-DR]. The POSTFIND Procedure and/or the DETAIL POSTFIND Procedure can be used to manipulate the corresponding record(s) if there is a need to. To change a value upon a Data Screen you would specify an ID number and then physically change the value. Once you go down to a specific field and change the value, then as in the Entry Phase, the ACCEPT verb is invoked and the same ten steps are executed. The only real difference in the Change Phase versus the Correction Phase, besides the processing mode being CHANGEMODE, instead of CORRECTMODE, is the ability to actually flag an existing record to be deleted in the upcoming Update Phase. To do this you would specify a "D" in the ACTION field or if working with DETAIL records on a screen by specifying "D-n/m" in the ACTION field. In both of these cases the DELETE Procedure is invoked along with its verb the DELETE verb. To physically delete this record you must invoke the UPDATE Procedure which contains the PUT verb which will interrogate the file's record buffer. The DELETE Procedure only flags the record for deletion. If this happens the status of the Record Buffer is changed to an ALTEREDRECORD and a DELETEDRECORD [-NR/ AR/ DR].

Next, as in the entry side, the Update Phase is invoked. The procedures of PREUPDATE, UPDATE, and POSTUPDATE will be executed if present. The PUT verb in the Update Procedure goes through its processing cycle again and performs the designated operation on the record depending on the status of the Record Buffer.

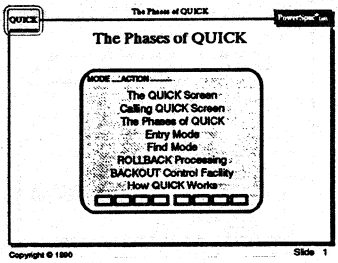
Another inherent feature of QUICK is the Backout Control Facility that is automatically invoked when record item fields are being passed back and forth on multiple screens This facility is used to maintain data integrity between multiple files when a file has been modified on one screen, and then passed to another screen, possibly declared as a Master file, and then a user backs out of the screen or data has been changed without doing an update. The information is normally written to a Backout Buffer in scenarios where files and/or TEMPORARY items are being used in this manner. In this case all files that were modified will be refreshed with the original values prior to the changes and backout.

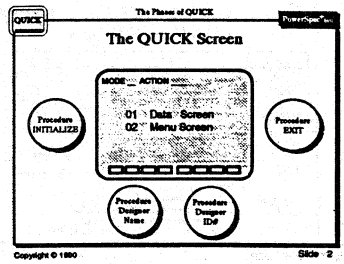
This is the flow of Find Mode. It has consisted of the Find Mode Initialization Phase, Path Phase, Selection Specification Phase, Data Retrieval and Data Phases, Change Phase, and finally the Update Phase.

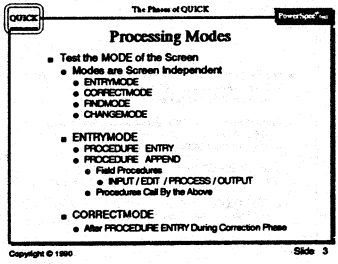
The Phases of QUICK are normally invoked by specifying some entry in the ACTION field. It is in these phases that the various procedures and their verbs are executed. The understanding of the timing of Processing Modes and changing of the Record Buffer Status is critical in writing procedures in QUICK. This paper has tried to give you a better understanding of How QUICK Works in the Phases of QUICK !

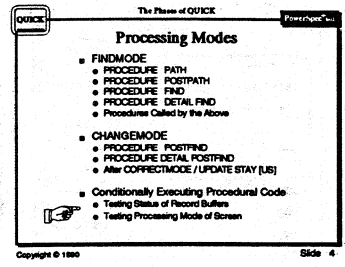
Good Luck !

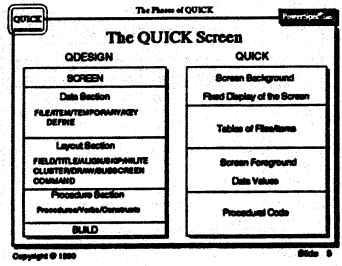
The Phases of QUICK

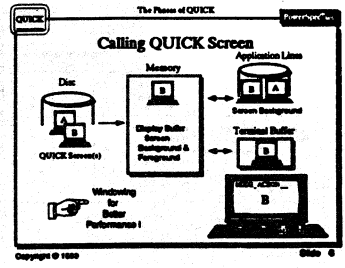




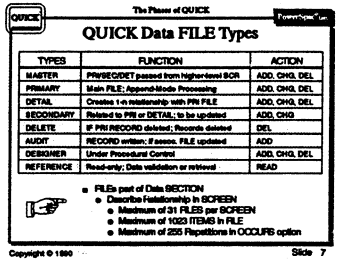


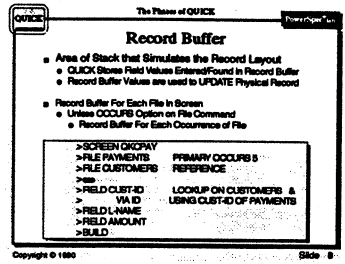


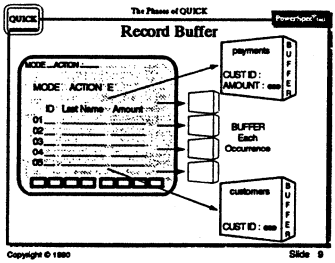


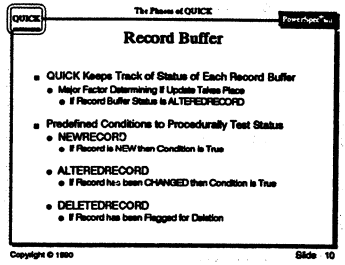


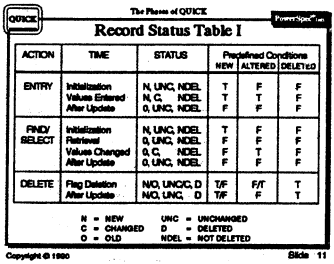
The Phases of QUICK

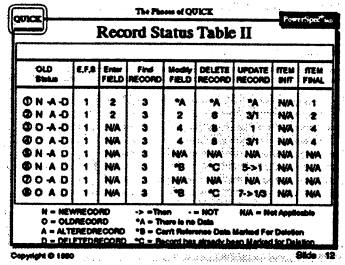






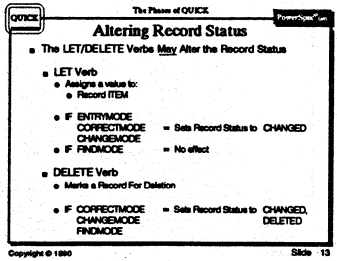


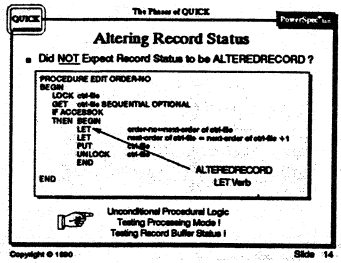


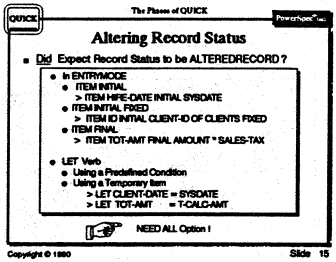


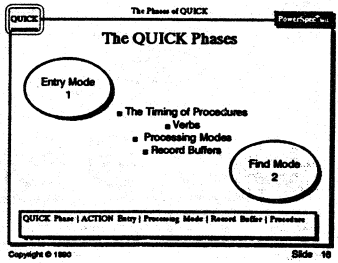
PowerSpec Intl.
"The PowerHouse® Specialists"
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

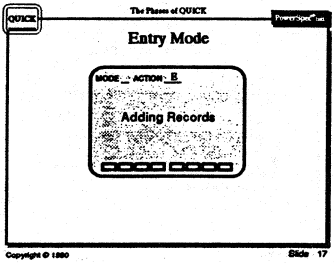
The Phases of QUICK

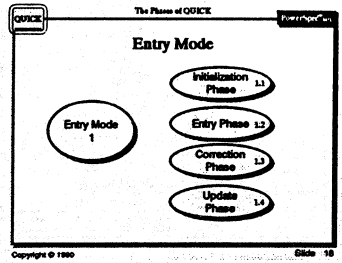






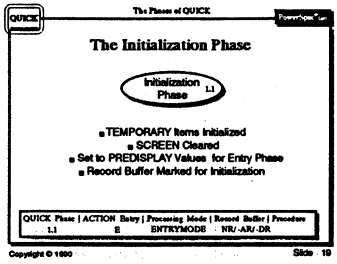


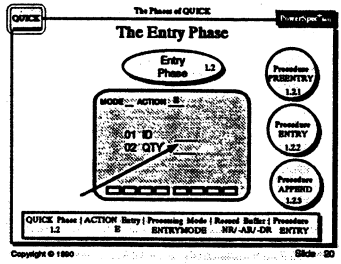


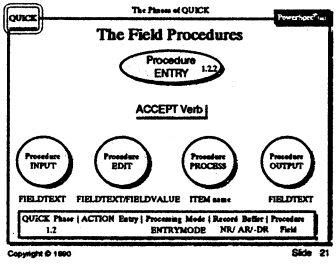


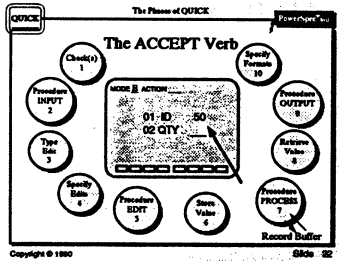
PowerSpec Intl.
"The PowerHouse® Specialists"
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

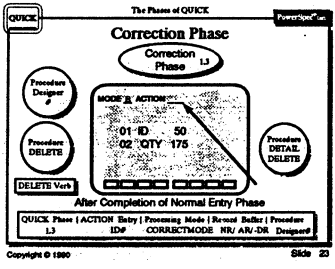
The Phases of QUICK

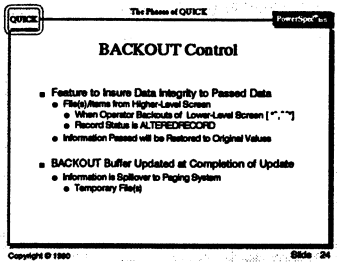






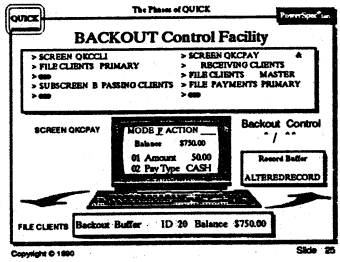


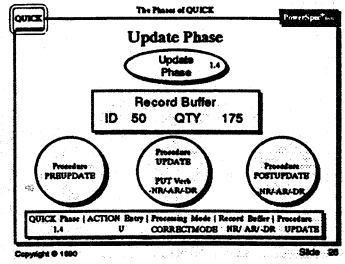


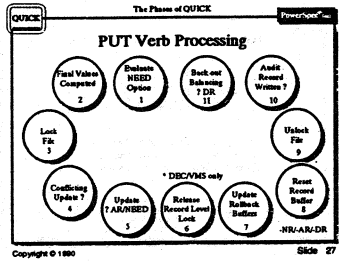


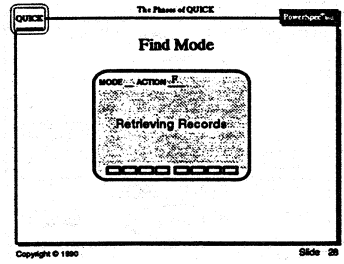
PowerSpec Intl.
"The PowerHouse® Specialists"
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

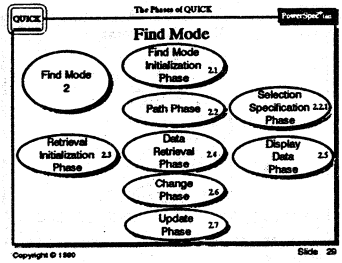
The Phases of QUICK

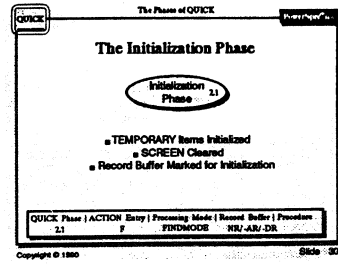






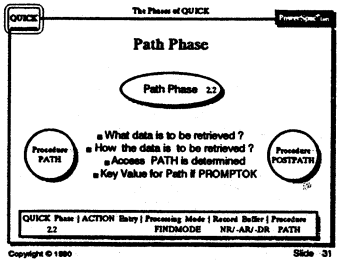


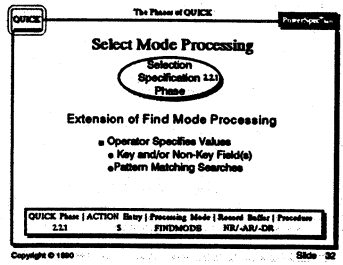


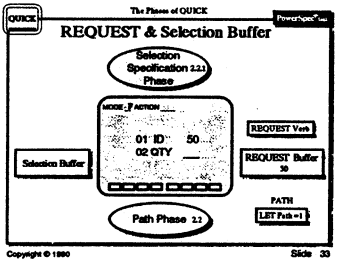


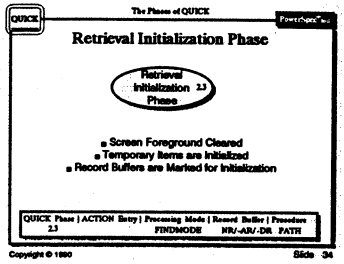
PowerSpec Intl.
" The PowerHouse® Specialists "
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

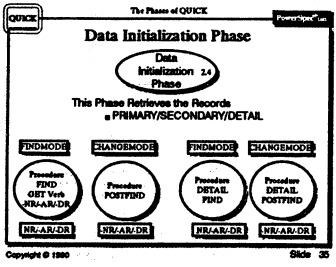
The Phases of QUICK

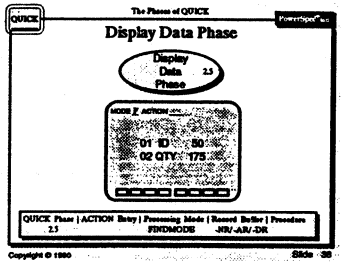






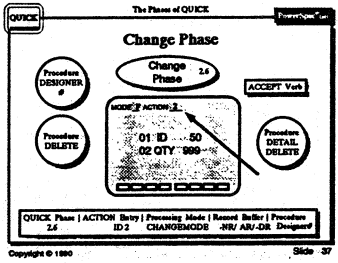


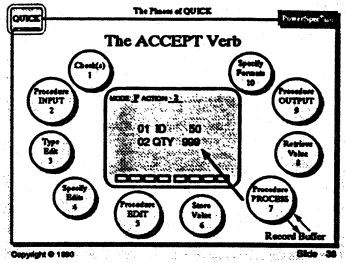


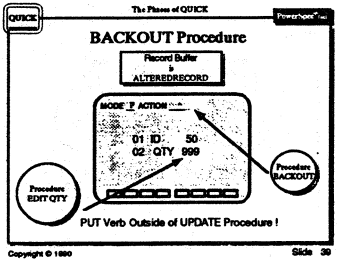


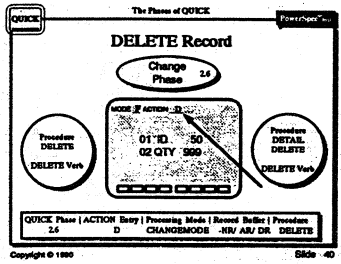
PowerSpec Intl.
" The PowerHouse® Specialists "
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

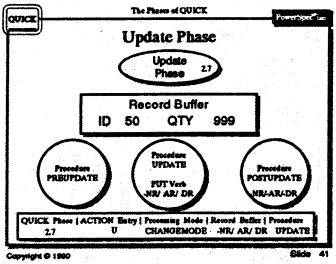
The Phases of QUICK

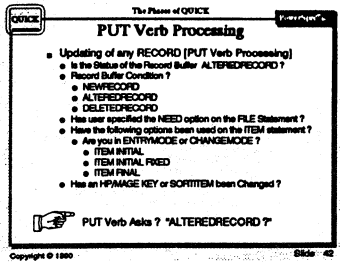






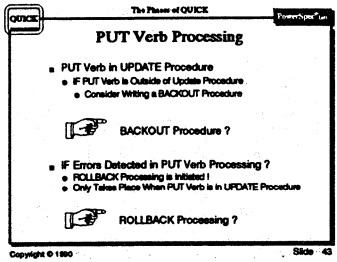


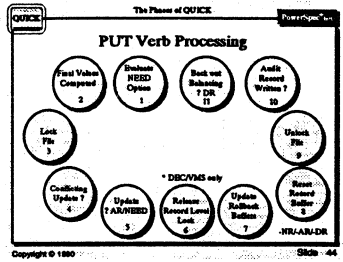


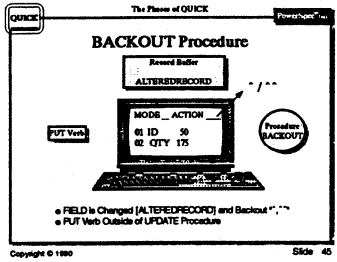


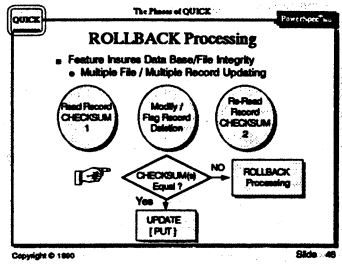
PowerSpec Intl.
"The PowerHouse® Specialists"
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

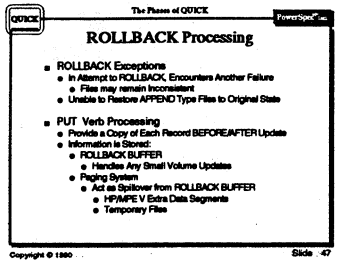
The Phases of QUICK

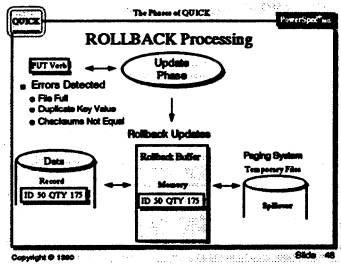






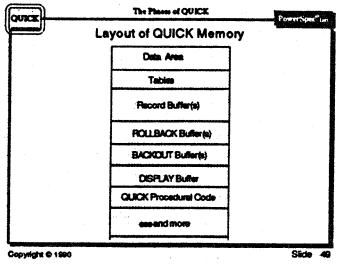


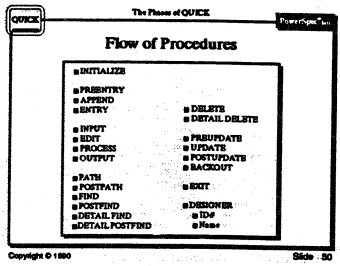




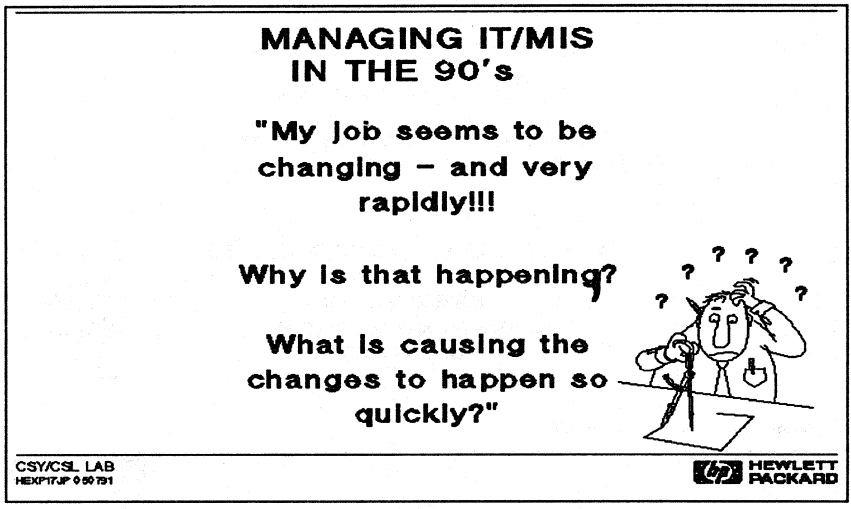
PowerSpec Intl.
" The PowerHouse® Specialists "
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA

The Phases of QUICK





PowerSpec Intl.
"The PowerHouse® Specialists"
403 Cross Lake Drive, Fuquay-Varina, NC 27526
Phone (800) 334-7023 FAX (919) 552-8061
USA



John Podkomorski
Hewlett Packard Company
19447 Pruneridge Ave
Cupertino, CA 95014

Information Technology and Management Information System (IT/MIS) management is becoming more complex as a direct result of industry forces, technology, and the changing expectations of management, users, and staff. These changes are often subtle, but have a very significant, and often confusing, effect on the day to day management requirements of the IT/MIS management team.

This session develops a model to help understand the changes which affect the IT/MIS environment. Historical pressures will be compared to new pressures for the decade of the 1990's. In addition, a group discussion will focus on possible strategies for managing the new set of pressures felt by IT/MIS managers.

AUDIENCE:

All levels of IT/MIS management, especially department management, and the management and staff of application development , systems design and implementation, and technology support.

MANAGING IT/MIS IN THE 90's

INFORMATION TECHNOLOGY LOOKING FORWARD TO CHANGES

"THERE ARE NO MORE TRENDS!!!

"EVERYTHING IS HAPPENING AT THE SAME TIME!!!"

Gedas Sakus, Northern Telecom 1991

CSY/CSL LAB
HEXPAIF.GAL_060791

HEWLETT PACKARD

FIGURE #1

The IT/MIS management world is becoming much more complex. Change has become so very rapid that it often feels as if the comfortable patterns of the past are all changing at the same time.

Where will IT/MIS managers feel pressure through the 1990's? How are the pressures changing? And why?

There are four basic sources of pressure on IT/MIS Management:

1. Upper Management
2. Users of IT/MIS services
3. IT/MIS Staff
4. Technological development, and industry trends.

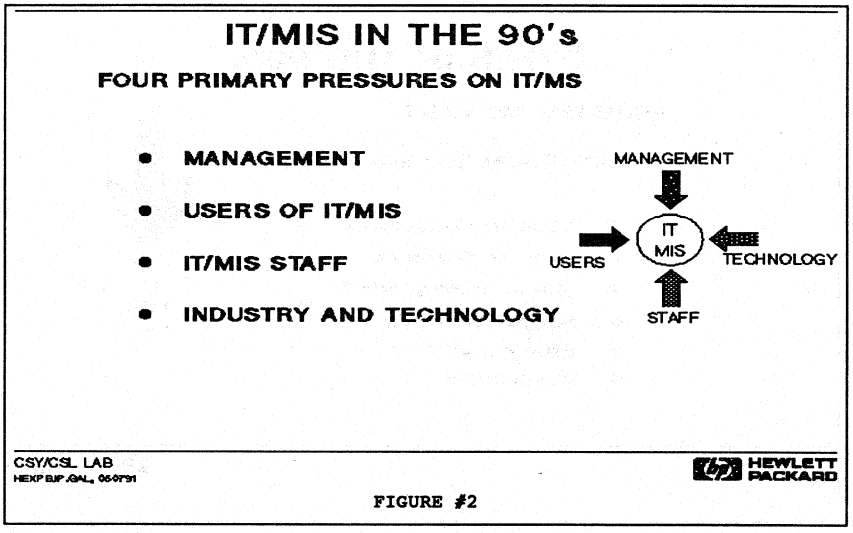


FIGURE #2

This paper will discuss the changes in each of these four very important areas.

This model is not really very creative. These four factors have been the major driving force behind IT/MIS for many years. What is changing, however, is the set of expectations each of the three organizational "users" now have for IT/MIS, and as we'll see later, the effect of industry developments and technology progress.

Let's start with Management. By management we mean the organization structure to which IT/MIS reports. Historically, this has been the Finance function in most companies, or perhaps the Manufacturing group, depending on the needs of each company. One change we see developing is the continued movement toward stand alone IT/MIS functions, at a par with other functions in the company. This change is one of the forces driving new sets of expectations.

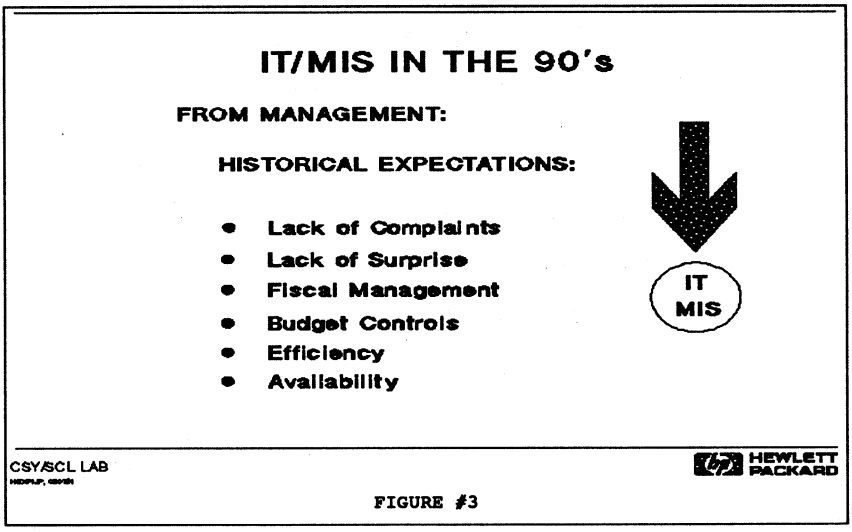


FIGURE #3

Historically, IT/MIS departments have been expected to be invisible, and have had the luxury of working in a limited monopoly situation. They had the technical experts, the equipment, the technology knowledge, and the endless jargon which kept the unruly users at bay.

Management expected the department to operate without complaints or surprises, and to manage their budgets with reasonable controls in place.

This expectation set has not changed significantly in the last 20 years or so.

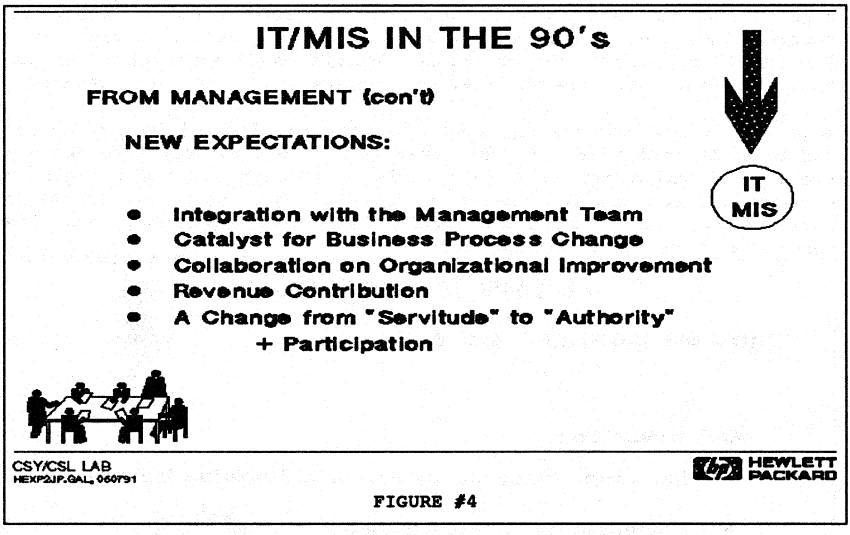


FIGURE #4

What's has changed?

The most significant change, which is made more visible by the movement to have separate IT/MIS functions in large companies, is the expectation that IT/MIS become more of a business partner with the other functions in the company. New demands are drawing MIS managers more into the management team to provide the "systems" expertise to solve business problems.

Information management has been perceived as a potential competitive advantage. Recent press articles, however, have begun to suggest that upper management is not satisfied with current MIS contributions.

Instead of a "servant" to the management team, IT/MIS is now being asked to be an active partner, with considerable emphasis on the IT/MIS contribution to organizational development, profitability, and business process improvement. IT/MIS authority seems to be increasing, with an absolutely necessary shift away from the comfortable ground of technology to the much more slippery arena of business factors. This often brings along an increase in authority for IT/MIS management, but not necessarily with the enthusiastic endorsement of the rest of the management team. The "politics" have changed.

There seems to be a current trend in industry emphasizing flatter organization charts with fewer layers of management. This style of

organization has a much different decision making model, especially compared to a deeper, more monolithic structure. There is a higher need for rapid, effective, and efficient information dissemination, as well as a requirement for flexibility, and responsiveness to rapid change.

Some IT/MIS environments may find the need for flexibility to be very difficult to work with. Fiscal pressure has continuously demanded cost reductions, which may have been achieved through increased rigidity, bureaucracy, and controls. Management today continues to ask IT/MIS for lower cost, but also wants more flexibility, options, and less bureaucracy, rigidity, and controls.

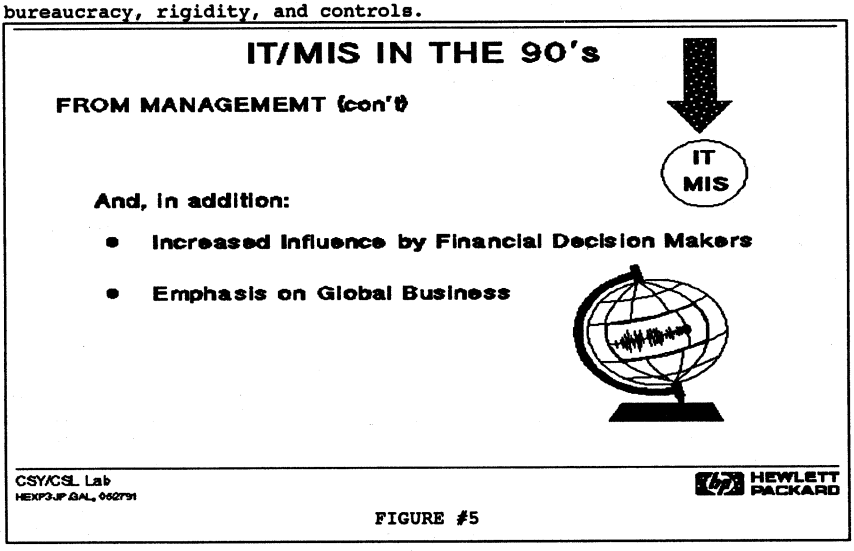


FIGURE #5

Another change in the dynamics of companies is the increased assertiveness of the finance function over technical decisions once tightly held in IT/MIS. Because of increased technical exposure, many more managers are becoming remarkably competent at asking very hard questions about the economics of IT/MIS development and operations. We have seen real cases where a change as radical as a shift in vendors has been caused by a directed budgetary limitation.

> "If finance says we must operate for $1M less per year, how can we do that? We may need to switch vendors!!".

Global influence is rapidly rising. Companies can be "national", "bi-national", "multi-national", or "global". Because of the shrinking size

of the world, many more companies are seeking opportunities to enter the global marketplace. The world market brings along many very interesting management challenges, and a whole plethora of technology challenges, which were not commonly considered even 10 years ago.

U.S. companies, in particular, are unaccustomed to the stresses and strains of international or global business. This new "global" requirement represents a new type of thought. Cultural issues, as well as the complexity of international finance, standards, and inter-country laws must be considered as an active part of "system implementation" decisions.

Here are some results from a recent survey done by the INDEX GROUP for 1991. The survey participants were a group of 394 U.S. and Canadian I/S executives, with companies greater than $250 million. 74% of these companies had revenues over $1 billion.

1991 INDEX GROUP SURVEY RESULTS

1991 RANK	DESCRIPTION	1990 RANK	1989 RANK	1988 RANK
1	Reshaping Business Processes through Information Technology	1	11	None
2	Aligning Information Systems and Corporate Goals	4	2	1
3	Instituting Cross Functional Systems	3	7	None
4	Boosting Software Development Productivity - developing systems faster, more cost-effectively	6	13	12
5	Utilizing Data so that information is accessible and used by the right person at the right time.	7	6	7
6	Developing an Information Systems strategic plan	5	4	2
7	Improving Software Development Quality	14	None	None
8	Creating an Information Systems Architecture	9	5	5
9	Integrating Information systems, solving technical problems of integrating heterogeneous h/w and s/w.	16	12	6
10	Enhancing Leadership Skills at or near the top of the IT/MISS organization.	None	None	None
11	Cutting IT/MIS costs	10	14	17
12	Using IT/MIS for competitive breakthroughs	8	1	4
13	Improving the IT/MIS human resource	11	8	8
14	Educating Management on IT/MIS	2	3	3
15	Connecting to customers and suppliers	19	None	None

1991 INDEX GROUP SURVEY RESULTS - continued

1991 RANK	DESCRIPTION	1990 RANK	1989 RANK	1988 RANK
16	Managing changes caused by information technology	17	10	9
17	Promoting the IT/MIS function	15	None	None
18	Determining the value of IT/MIS	23	20	None
19	Managing dispersed Systems	25	16	13
20	Capitalizing on advances in technology	21	17	None

As you can see, the list of top 20 needs of the IT/MIS have shifted a little over the last four years, with management needs expanding, and technology needs dropping a bit.

Let's shift, now, and look at how the expectations of USERS of IT/MIS are changing.

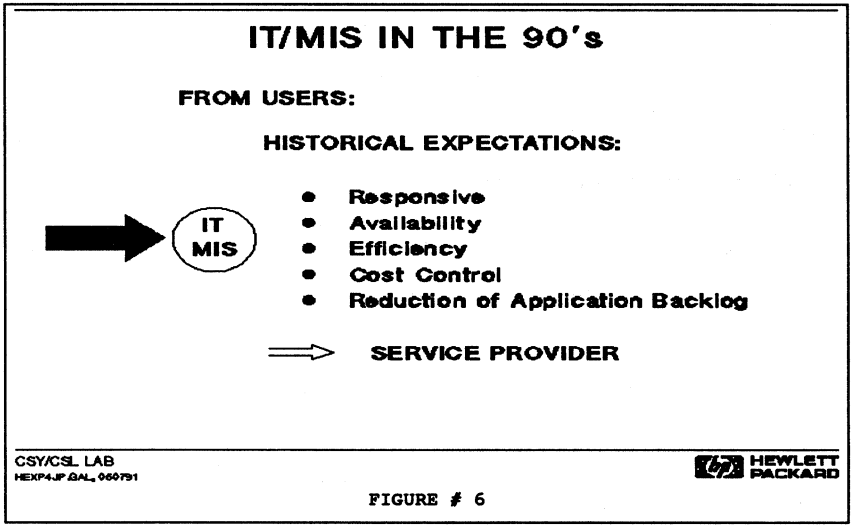


FIGURE # 6

"Users", in this context, are the departments in a company who actually receive the benefit of IT/MIS development and operations work.

Historically, users expected solid, reliable, available, cost effective services from IT/MIS. This too has not really changed in 20 years. Users have always been concerned about their application development backlog, but have grudgingly accepted it because of the perceived difficulty of implementation of complex IT/MIS programming projects.

IT/MIS provided a service, crunching numbers and inventory records, and keeping track of historical data better than people could.

How are users' expectations changing?

In historical IT/MIS installations, user personnel would operate a single application, often heads-down to the keyboard, for hours on end. They were specially trained for that application, and worked with it for a long time. Specific application functions needed to be well tuned, but users did not need to be very versatile, and it was acceptable to use training to overcome some system shortcomings.

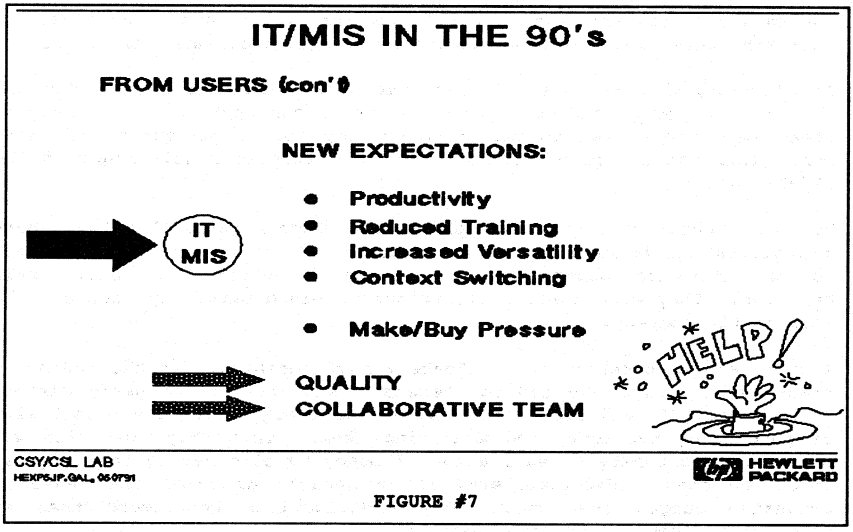


FIGURE #7

User departments are drowning in the combination of budget restrictions, personnel shortages, labor shortages in key skills, and an increasingly demanding business environment.

Because of cost pressures, there is a very significant emphasis on productivity in user departments. User personnel are being asked to operate efficiently in multiple applications. User departments expect their staff to be multifaceted, and fully knowledgeable in multiple, unrelated, applications like word processing, payroll, customer management, phone inquiries, and electronic mail.

There is a dramatically increasing need for user proficiency, with a corresponding desire for less required training. Modern training is expected to be painless, blindingly quick, and to be self-paced, without a required instructor.

The productivity of user personnel is now affected not just by training, or by experience, but by the complexity of the multiple application environments, and by the time it takes to switch contexts between them. One HP customer calculated that context switching alone - the time to switch from one application to another - was costing over $2M each year.

Users read every day about decreased training time, enhanced context sensitive help facilities, intuitive user interfaces, and "push button"

instantaneous context switches, and they are rapidly expecting IT/MIS to implement these more powerful environments into their everyday lives.

In addition, more computer literate users are demanding a much stronger role in the design and development of their own applications. Many of these same users feel empowered to develop some applications on their own. This pattern places new training and support requirements on the IT/MIS function.

Users no longer want turnkey applications which require that they learn the quirks and foibles of the application. They want a user interface to be knowledge based, intuitive, and adjustable to match their behavior. They want their applications to blend naturally into all the work of the company.

Another subtle pressure is the "make / buy" decision. In the '60's, we "made" everything. We had to, because there was no 3rd party market. In the late 70's and early 80's, IT/MIS began suggesting the possibility of purchasing packages, and modifying them. This "buy" decision was often an opportunity to save a lot of money by eliminating the original invention cost. Unfortunately, it was nearly as often the cause of extensive budget over runs, when modifications cost more than an original implementation might have.

Times have changed with the development of a very viable 3rd party market. There are many more applications on the market, many of which are very directly applicable to common business problems.

Our users have adopted the perception that "buy" will work - "It says so right here in the paper, so it must be true". This puts IT/MIS in a very delicate position. If IT/MIS suggests a "make" decision, users ask why it can't be bought cheaper. If IT/MIS suggests a "buy" decision, users ask why we need such a big IT/MIS department for, and "Is it installed yet and running exactly right, and why not?"

In addition to all this, users are fully expecting unprecedented quality levels - much higher than in the past. "Quality" has been talked about, and written about, a great deal in the press, and in professional circles. Users now expect very few problems with new systems or applications and are very intolerant of down time or interruptions. As more users become technically "computer literate", this expectation will very likely increase. Budget and staff reductions in user organizations make outages much more expensive to the users, and any data loss is critical.

IT/MIS is increasingly expected to leave its comfortable role as a service provider, and enter a newer, more complex collaborative team player. This newer role requires much more than IT/MIS technology

skills; organizational development, finance, business judgment, and customer management are increasingly important.

Let's look at how staffing problems are changing.

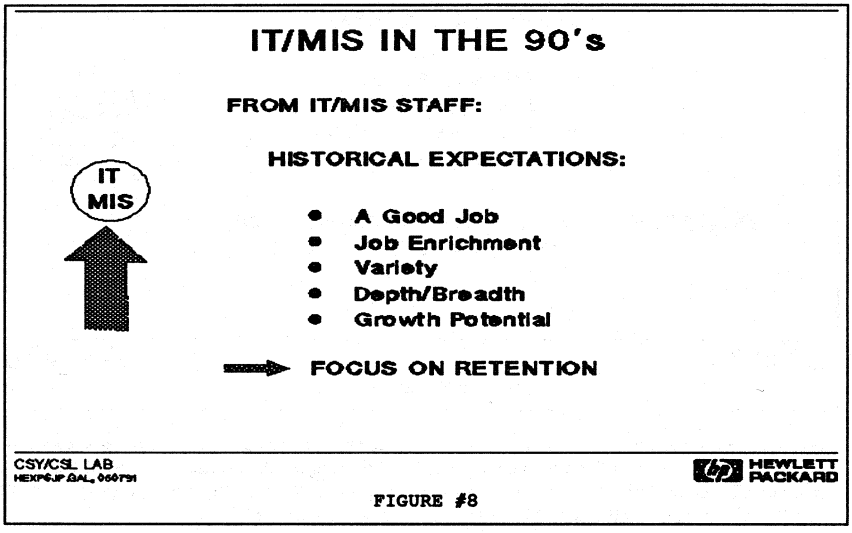


FIGURE #8

IT/MIS staff have always been an unruly lot. Because they have had special knowledge they have demanded, and have usually received, some special treatment.

Historically, staff has expected a "good" job, with lots of variety, job enrichment opportunities, the ability to develop depth or breadth (personal preference), and growth potential.

Programmer analysts were competent if they had a good working understanding of COBOL, Screen management, and a data base. Specifications were usually technically developed, and given to programmers to implement.

The historical focus on staff management was retention of your best employees. In many cases, recruiting could be easy, depending on your environment and how much you were willing to pay.

What's changed here?

IT/MIS IN THE 90's

FROM STAFF:

NEW EXPECTATIONS:

- **State of the Art**
- **Development Environment**
- **Increased Training**
- **Skills Development**

FOCUS ON RECRUITING OF VERY RARE RESOURCES

CSY/CSL LAB
HEXPTJF.GAL, 060791

HEWLETT PACKARD

FIGURE #9

The major change has been the market's reaction to the decreasing numbers of qualified technical staff. Employment focus, in addition to retention, must now be on recruiting requirements. In some ways, IT/MIS management must continuously recruit its current employees.

In the past, personal progress could have been measured by promotion, or by technical growth. The promotion path is not as available today, because of a trend toward flatter organizations, and toward less staff growth. There are simply less managers, so there are less chances for promotion.

On the technical growth side, developers, programmers, programmer analysts now demand to work "at the state of the art". They ask for strong, nurturing environments, with superb tool sets, increased training, and skills development plans.

Tomorrow's technical staff will require much more than language, terminal, and data base knowledge. They will need sound business judgment, consultative skills, and a broad knowledge of all the aspects of the business enterprise. With user departments capable of doing some of their own programming, the type of technical contribution made by IT/MIS staff will change a lot in the near future.

This feels quite similar to the needs of the past, but is different in one very important way. The decreasing numbers of qualified job applicants is making the job environment very competitive.

A different type of management style is increasingly required to attract, manage, and retain qualified technical talent. More focus on individual needs, job enrichment, personal growth, and the whole working environment is required today.

Finally, let's look at the effect of industry and technology changes.

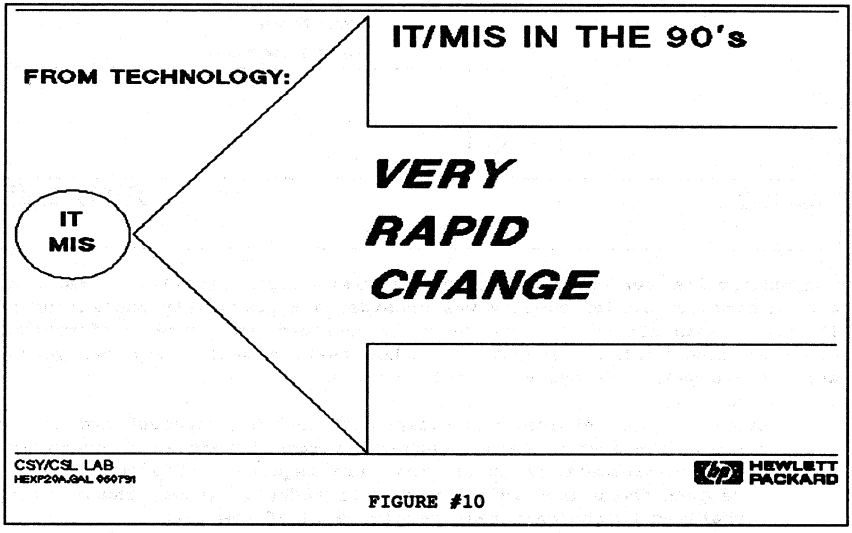


FIGURE #10

This is the area which is the strongest contributor to the rapid change in the IT/MIS environment. Technology change is happening at an unprecedented pace, throughout the industry. At the same time that technology is changing rapidly, we've seen that the forces of business changes are moving equally rapidly. The combination of these two factors amplifies the effects of both.

Let's look at just a few of the industry issues IT/MIS needs to deal with.

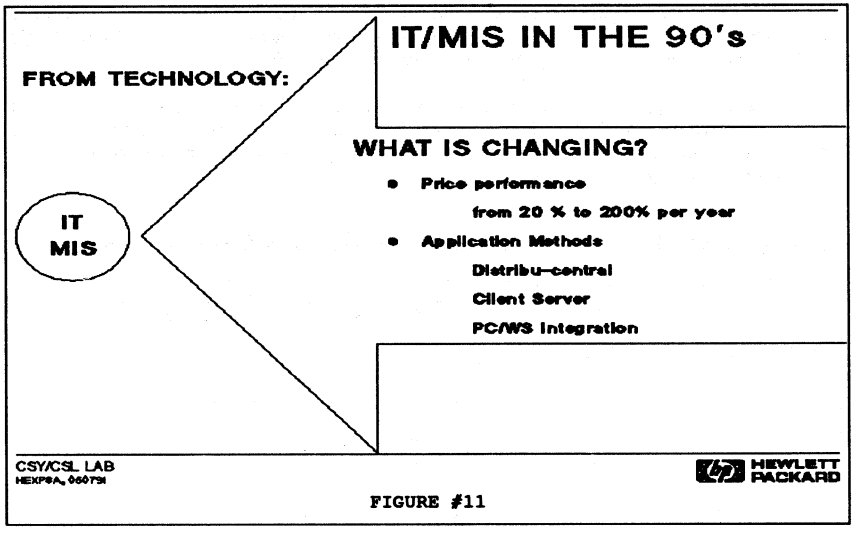


FIGURE #11

Computer price performance curves are rising dramatically. Some time ago, a computer vendor company was considered aggressively successful if it could maintain a 20% per year improvement in price performance. Computer models were released to market perhaps once every two years, and a 5 - 8 year life cycle was not uncommon.

The historical gaps between "mini-computers" and "mainframes" are not as clear today. "Mini-computers" supporting many hundreds of users are common. The management of these computers requires different resource algorithms than those used in the past. In today's market, there really are alternatives to the mainframe requirements of the past.

Today's vendors are progressing at nearly 200% per year. It is not unusual to see very rapid announcements of new computer lines which totally replace the "state of the art" just 9-12 months back. Purchase decisions are increasingly difficult as a result. IT/MIS must ask two questions, and balance the answers carefully...

> What if I buy today, and a newer, faster, cheaper model is released in 3 months? I'll really look silly!!!
>
> What if I don't buy today, and my need for increased computing causes a business problem? I'll really look silly!!!.

Another contributing factor is the development of new technology in application development methodology. Increased power in PC's and workstations, coupled with the software technology to utilize that power for user interface enhancements, and for file and data storage distribution is really changing the profiles of what new applications look like. The definition of "central" computing, or "distributed" computing, is getting pretty fuzzy.

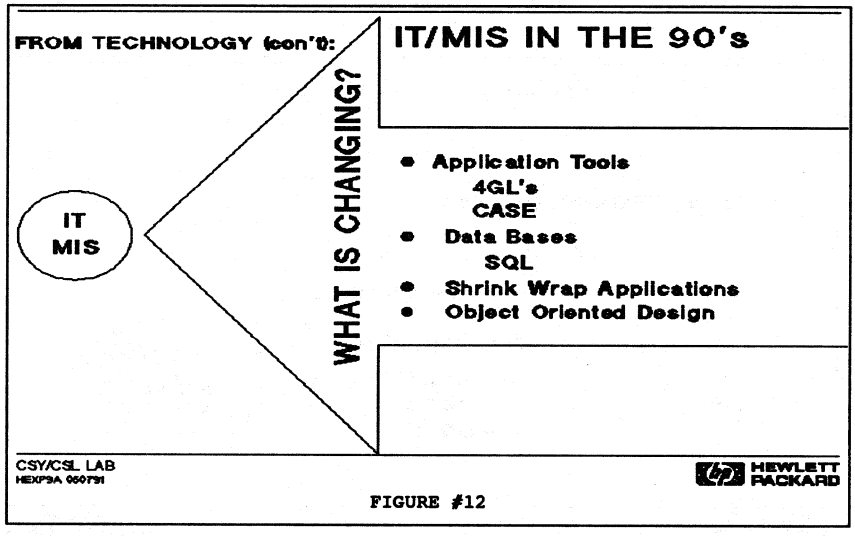


FIGURE #12

The recent entry of computer engineering (CASE) products into the marketplace is changing the expectations of IT/MIS staffs, as well as those of our users. CASE tools are rumored to make development much faster, and with "perfect" results. Developers and designers are beginning to demand access to CASE technology (perhaps in self-defense) in order to increase their own productivity.

4GL's have also increased our user's expectations about the speed with which applications can be delivered. 4GL's promise development times considerably lower than historical methods.

4GL's have another interesting effect. As computer literacy increases in the user base, more and more users are feeling competent in inventing or modifying their own applications. IT/MIS has considerably less control in an environment where users can write inquiries, reports, and even update their own records using 4GL technology.

The advent of relational data bases, and SQL access methods, amplifies this trend. As the elements of a company's data become easier to manage, and more intuitive to access, the nature of system design and implementation must change to allow for the considerably higher level of user freedom currently available. "Object" orientation is challenging the technology limits again.

Also, development of software packages - "shrink wrapped" software - is increasing our users' appetite for solutions.

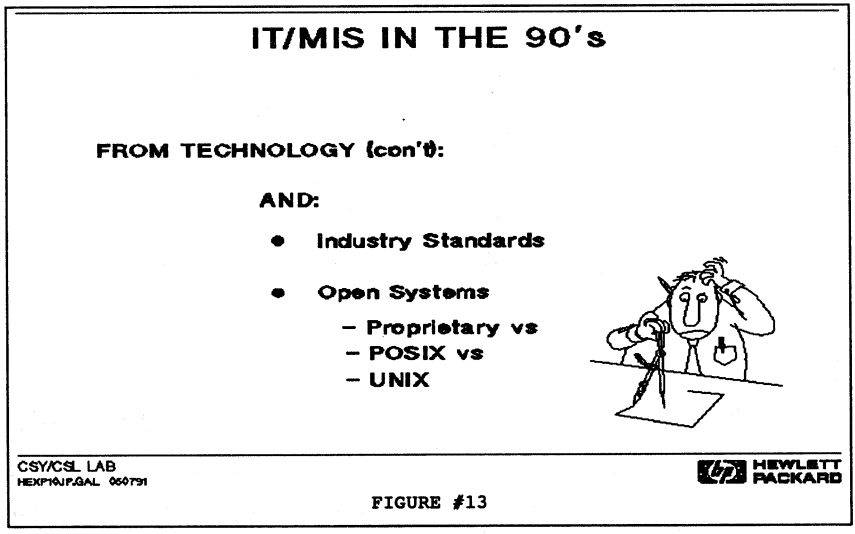


FIGURE #13

And then there is the "open systems" question. What is the "right" thing to do in an IT/MIS department? How do we decide whether to make changes across vendor boundaries, or Operating System boundaries? What criteria should we use? If we stay proprietary, will that mean we're "trapped"?

Will our staff continue to work for us if we don't go to UNIX? Will our staff continue to work for us if we do go to UNIX?

Tough decisions everywhere...

IT/MIS IN THE 90's

FROM TECHNOLOGY (con't):

INDUSTRY HYPE!!!

- **We vendors are a very creative lot!!!**

CSY/CSL LAB
HEXPTUP.GAL 060791

HEWLETT PACKARD

FIGURE #14

Which is why you experience all the HYPE!!

We vendors are a very creative lot, and will find many ways to present each technical idea, product, or fantasy in a way which makes it sound like the very best thing that ever happened to the world.

Which brings us to the IT/MIS situation for the 90's. Pressure, pressure everywhere, and not much relief in site.

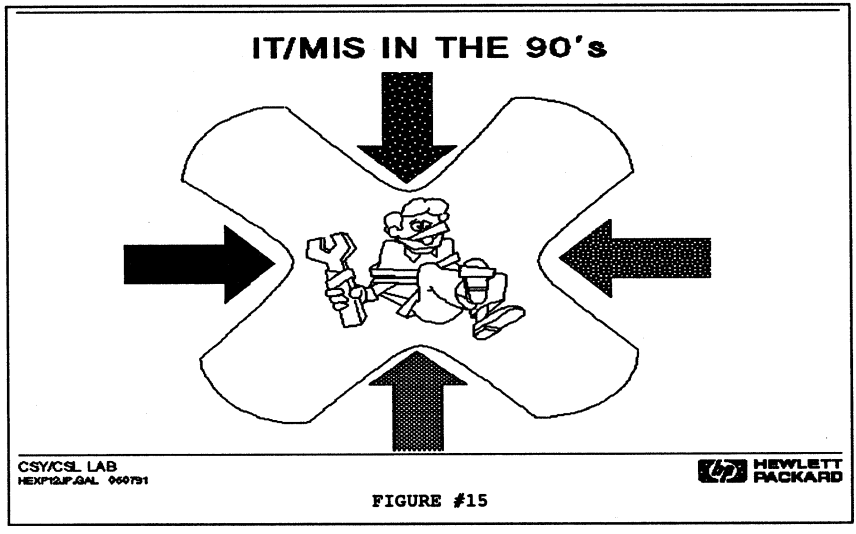


FIGURE #15

IT/MIS management is increasingly expected to play a stronger role in the corporate management team, making a direct contribution to the business.

IT/MIS management is increasingly expected to collaborate with its users, no longer providing services, but pro-actively improving productivity in a rapidly changing, complex application environment.

IT/MIS management must react to a growing shortage of technical staff, with the resulting need to really focus on staffing issues.

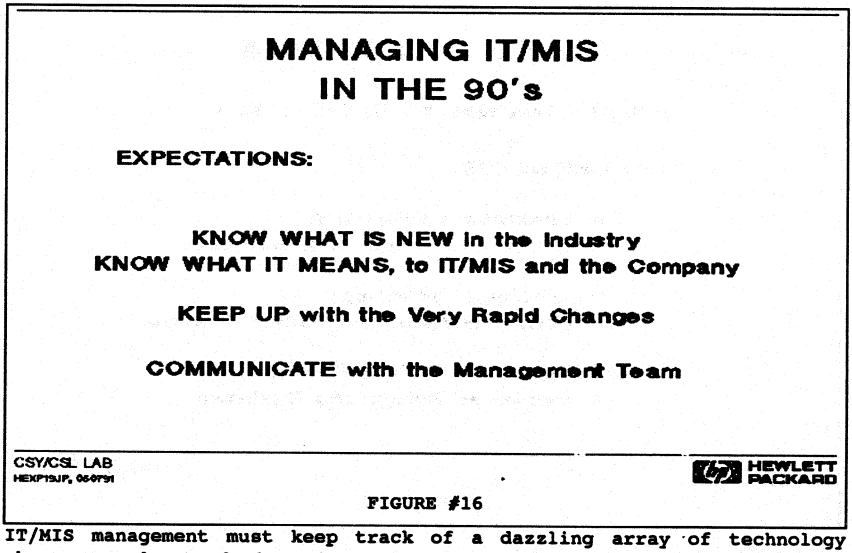


FIGURE #16

IT/MIS management must keep track of a dazzling array of technology changes, understand what they really mean to the company, measured in business terms, and become very effective at sharing that knowledge with the rest of the company.

More communications skills are required today than have ever been required in the past.

More business skills, and business systems skills are required than have ever been necessary before.

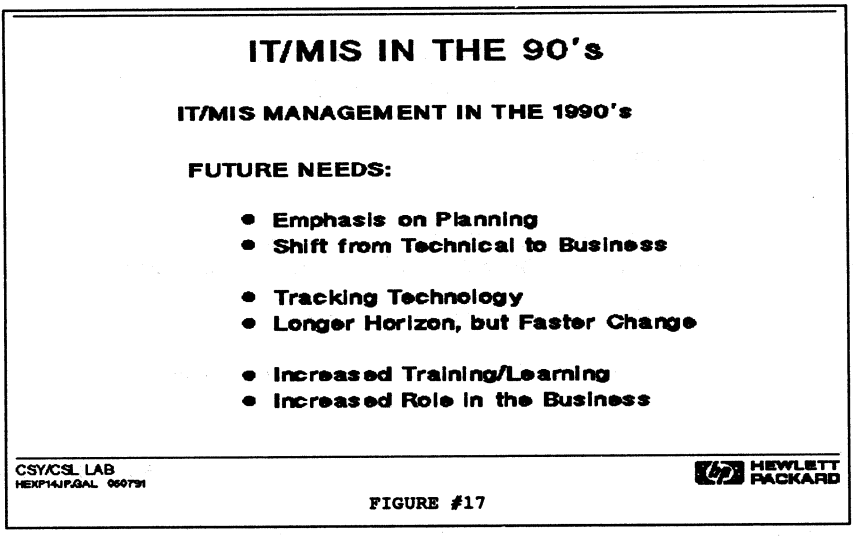


FIGURE #17

In summary, IT/MIS must look forward, and develop new skills very quickly, to deliver on the expectations of upper management.

A shift from technology management to business management, and a new way of contributing to the management team, is increasingly important.

Tracking technological change is probably the largest current challenge. Technological change is happening so quickly that it is very difficult to keep track, or to make decisions. Management wants decisions made on a "long term" basis, with predictable stable results. The technology world seems to be changing under our feet about every 9 months or so.

IT/MIS will need to do more training of its users, and will need to shift its role from service provider to technology mentor. Controls on user data will be minimized, and access to data optimized.

And all this must be done in chronically tight fiscal situations, with an increasing shortage of qualified technical personnel.

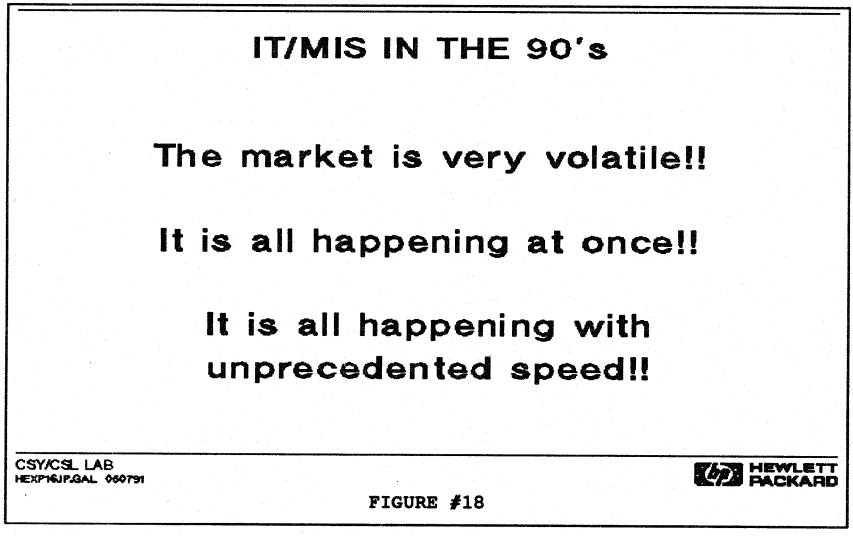


FIGURE #18

The computer marketplace is incredibly volatile. Change is everywhere.

It is all happening at once, on many different axes, in many product areas, with global implications.

It is all happening with unprecedented speed.

And we are all in the middle of it!!! What an exciting time!!!